Identity-Focused Curriculum Design and Review:

A Workbook for Liberal Arts Computer Science

{Authors anonymized during SIGCSE Review}

Introduction

Curricular guidelines, such as the CS2023: ACM/IEEE-CS/AAAI Computer Science Curricula that is currently under development, can be a valuable resource for those creating and revising curricula. However, the guidelines are intended to address computing curricula in computer science and closely aligned fields at a wide variety of institutions and can be overwhelming to consider. There is no "one size fits all" for computing curricula, given the variety of contexts in which computing programs operate. In particular, liberal arts curricula often leverage their unique institutional context and mission to create innovative programs. Indeed, it is not practical, and most likely not useful, to blindly consider a curricular guidelines document like CS2023 and decide to adopt or modify a program at a particular institution without a clear understanding of the program's opportunities and constraints. It is important to develop, or at least clarify, the institutional and departmental identity and goals, and view the curricular standards through that perspective.

This workbook is designed to help faculty, primarily in computing departments that operate within a liberal arts context or otherwise reflect a liberal arts perspective, work through their program's current and desired identity and goals to develop (or improve) a set of design principles and program level learning outcomes before turning to CS2023 to inform curricular re-design. Institutions that adopt a liberal arts perspective, "pursue a philosophy of higher education that emphasizes preparing students for the full range of thinking they will face throughout their lives; thinking in the service of a career, thinking in order to participate in civic affairs and society generally, thinking in order to have a fulfilling personal life, etc."[1]. While the approach outlined in this workbook can be applied to all computing programs, it emphasizes priorities that tend to be most important in a liberal arts context. These include strong ties to an institutional mission, an emphasis on a broad educational experience, and the constraints imposed and opportunities provided that result in unique and innovative curricula that exist within that context. Further motivation for this process and the research behind its design is available in a companion article by the authors.[2]

---

[1] Douglas Baldwin, Amanda Holland-Minkley, and Grant Braught. 2019. Report of the SIGCSE committee on computing education in liberal arts colleges. ACM Inroads 10, 2 (June 2019), 22–29. https://doi.org/10.1145/3314027

[2] In submission: *Computer Science Curriculum Guidelines: A New Liberal Arts Perspective*

How to use this document

Just as a curricular guidelines document is not one size fits all, this workbook can be used in different ways. Some departments might have an outdated curriculum, developed by a group of faculty who are no longer there, or an outdated focus based on the origins of the program in either engineering or mathematics. Such departments might spend significant time on the first few steps to get a good understanding of the program's current identity and goals, articulating what the faculty would like their identity and goals to be (if different), and how current and future departmental identity and goals align with institutional identity and goals. Other departments might be coming off of a periodic program review and have just worked through many of the steps, and would be prepared to start looking at the final steps almost immediately. Some may be looking to make more incremental changes to their curriculum, while others may be designing a curriculum for a new program.

These different goals will affect how you use the workbook. A good way to start is to read through the entire workbook, and decide which parts would be most beneficial in the short term. In some cases, the process could be fairly quick, while in others it could require months or even a few years to work through. For a comprehensive program revision, it might not be desirable or possible to work through the entire process in a short period of time and starting with setting semesterly or yearly goals could be helpful. Programs just wishing to do a periodic review of their curriculum or who come to the process with documentation from previous curriculum design work may be able to work through the process more efficiently.

The workbook could be used as a framework for at least part of required periodic institutional program reviews for departments in institutions that require them, or as a standalone exercise to inform departmental planning. Even if not at the point where such a review is due, working through this process can make the next such review more straightforward. Departments planning to hire or wanting to make the case for additional faculty may find the process useful for identifying areas in most need of coverage and providing supporting material for a position request.[3]

A subset of faculty could be designated to lead the primary work for this process, but it is important to collect feedback and build consensus among the entire department at each step. It is also important to seek buy-in and support from other stakeholders on campus. Points in the process where this is especially important are noted throughout this document.

---

[3] James D. Teresco, Andrea Tartaro, Amanda Holland-Minkley, Grant Braught, Jakob Barnard, and Douglas Baldwin. 2022. CS Curricular Innovations with a Liberal Arts Philosophy. In Proceedings of the 53rd ACM Technical Symposium on Computer Science Education (Providence, RI, USA) (SIGCSE 2022). Association for Computing Machinery, New York, NY, USA, 537–543. https://doi.org/10.1145/3478431.3499329

Prerequisites

Users of this workbook should have familiarity with the process of developing curriculum at least at the course level. This includes an understanding of developing learning goals, assessments, and activities. We recommend L. Dee Fink's *Self-Directed Guide to Creating Significant Learning Experiences* [4] for an overview of these concepts.

Licensing:

[4] L. Dee Fink. 2013. Creating Significant Learning Experiences: An Integrated Approach to Designing College Courses. John Wiley & Sons, Hoboken, NJ, USA. Google-Books-ID: cehvAAAAQBAJ. https://www.bu.edu/sph/files/2014/03/www.deefinkandassociates.com_GuidetoCourseDesignAug05.pdf

Workbook

Step One: Articulating Your Identity

As a first step you will reflect on the unique identity of your program as a computer science program within the liberal arts. This will include institutional elements as well as elements defined by your department and its faculty. This step will have you collect relevant information about your identity and work to come to a consensus statement of your view of the liberal arts and your identity as a liberal arts computer science program.

Institutional Level

Identify and review any of the following that exist at your institution, considering the institution as a whole, Academic Affairs, and/or the College/Division in which you are located. Similar items that are department/program-specific will be considered in the next step.

- Mission Statement
- Vision Statement
- Values Statement
- Statement on Diversity, Inclusion, Equity, and Access (DEIA)
- Statement of Institutional Student Learning Outcomes
- Honor Code or Academic Integrity Policy
- General education, graduation requirements, and educational goals and outcomes: consider how both catalog language and internal documents articulate their purpose
- Strategic Plan Documentation
- Fundraising/Capital Campaign Objectives
- Admissions/Marketing Materials
- Faculty Review Criteria

1.1. Taking these resources together, describe your institution's particular vision for, or interpretation of, liberal arts education. Be specific about what is most important or distinctive.

Departmental Level

Identify any of the following that exist for your department or program:

- Mission Statement
- Vision Statement
- Core Values Statement
- Diversity, Inclusion, Equity, and Access (DEIA) Statement
- Current programmatic student learning outcomes
- Departmental Strategic Plan
- Departmental External Review Documentation
- Catalog Description of Program
- Admissions and Marketing Description of Program
- Faculty Review Criteria
- Recent position descriptions/job ads
- Strengths and interests of current faculty
- Processes and practices with respect to DEIA

If you do not have existing Mission, Vision, and/or Core Values statements, consider creating these before proceeding. A visioning toolkit is available online at: https://atctools.org/toolkit_tool/visioning-toolkit/

If you do not have existing programmatic student learning outcomes, you can create them as part of a curriculum revision process as you progress through this workbook (See Step 3).

1.2. Taking these resources together, describe your department's particular vision for or interpretation of computer science education. Describe carefully how it connects to or reflects its liberal arts, institutional, and departmental contexts.

Identity Statement

1.3. Looking at the connection between your institutional identity (1.1) and your departmental vision and interpretation of computer science (1.2), develop a statement of identity for your program as a liberal arts computer science program. This may take the form of new or revised mission, vision, and/or core values statements for your program.

In the following steps, this statement will be used to articulate your curricular values and goals and drive decision making, so you should have strong departmental consensus around this statement and may wish to obtain administrative buy-in as well.

Step Two: Stating Your Design Principles

After articulating the mission for your program, identify design principles that will shape the structure of your curriculum. Here we draw on what Fink[5] calls “Situational Factors” in Step One of his process (identifying the teaching/learning context, nature of the material, and characteristics of the students and instructors; refer to the questions on page 7 for additional guidance). We also draw on our own disciplinary background in requirements analysis and the design process, applied now to curriculum rather than to software. We again emphasize an approach that considers your institutional and department identity. Below we divide these principles into two categories: identity-based principles focused on the identity you described in Step 1, and structural design principles based on institutional constraints. We note that there is some overlap between identity-driven and structural design principles. For example, your identity may motivate a desire to contribute broadly to institutional general education requirements, or you may have a structural need to do so either because it is required or a necessity for enrollment. We have included a number of examples of identity and structural design principles in Appendix A, but do not feel limited by these lists. We then draw your attention to reflecting how DEIA is represented in your design principles because of the importance of providing education that is inclusive, equitable, and accessible to diverse students.

Identity-Driven Design Principles

Review your identity statement from Step 1.3. Ask yourself what structure or broad elements for a curriculum would be most important to ensure the student experience is strongly aligned with your identity.

For example, if your institution has a mission of recruiting and educating first-generation students, and your department similarly focuses on supporting this student population, you should have one or more design principles related to this goal. Depending on how well developed this portion of your identity is, it might be as simple as “supports the needs of first generation undergraduate students.” Or you may articulate your understanding of those needs within your student population in more detail, perhaps as relates to entry points into your major, flexibility, and mentoring.

---

[5] L. Dee Fink. 2013. Creating Significant Learning Experiences: An Integrated Approach to Designing College Courses. John Wiley & Sons, Hoboken, NJ, USA. Google-Books-ID: cehvAAAAQBAJ. https://www.bu.edu/sph/files/2014/03/www.deefinkandassociates.com_GuidetoCourseDesignAug05.pdf

While doing this, be careful not to describe what your curriculum *already does*. Instead, reflect on aspirational design principles driven by your identity. Subsequent steps will allow you to take into account your current state and operational constraints.

2.1. What are your identity-driven design principles?

Structural Design Principles

In addition to the design principles that come out of your identity, it is important to understand the practical requirements and constraints that your program operates under. As described above, these may come from institutional requirements on majors or be shaped by department resources. Consider if your structural design principles can be phrased in neutral terms or as opportunities, rather than as limitations.

As you proceed with this step, think carefully about any requirements or constraints that might come out of department tradition or history but could now be revisited. For example, while a three-person department is unlikely to become a ten-person department, this is a good opportunity to consider whether you must offer a curriculum that can be staffed by no more than three full-time faculty (which perhaps you do), or whether your institution would consider adding staffing in order to support an updated curriculum.

If you haven't done this recently, you should also confirm that you have an up-to-date understanding of institutional requirements for majors. You may want to review guidance or forms that your curriculum committee or academic affairs require when revising an academic program.

As you complete this step, be careful to be complete but not over-constrain your curriculum design.

2.2. What are your structural-design principles?

DEIA Design Principles

Diversity, Inclusion, Equity, and Access are no longer optional considerations in curricular design. The computer science community has broadly adopted a commitment to providing equitable and accessible education for all students. This is reflected in the ACM/IEEE Computing Curricula 2020 (CC2020) Report as part of its vision for curricular guidelines in all areas of computing for the coming decade. In that report's introduction, they say:

> *"One underlying theme of the Report is the development of computing talent from all sectors and groups in our society. A lack of diversity limits creativity and productivity. It excludes many potentially qualified individuals and can be a significant concern for many employers. For example, the underrepresentation of women in computing in some countries has received much attention [Reg1]. This Report recognizes the importance of diversity and recommends that academic computing departments promote best practices to attract and retain greater student diversity."*[6]

Reflect on your institutional context, including the particular DEIA challenges you face, goals that have been set, or initiatives that are underway. Think additionally about your program context and the ways it is similar or different from your institutional context when it comes to DEIA. If you are new to this work, consider seeking out support from an Office or Director of DEIA on your campus. NCWIT also provides helpful resources for higher education: https://ncwit.org/higher-ed/

Review your identity-driven and structural design principles and ask yourself if they encompass principles that will help you work towards greater diversity, inclusion, equity, and accessibility within your program and its curriculum. Consider if there are additional design principles you want to add to support this work.

2.3. Which of your design principles will advance diversity, equity and inclusion efforts on your campus and in the discipline of computer science?

[6] https://dl.acm.org/doi/book/10.1145/3467967

Step Three: Articulate Program-Level Learning Outcomes

Programmatic student learning outcomes actualize the vision set out by your identity and your design principles by describing what particular computer science knowledge and skills your program considers most essential. A useful resource defining student learning outcomes is available at: https://www.colorado.edu/oda/sites/default/files/attached-files/program_learning_outcomes_v2.pdf.

If you already have program-level student learning outcomes, you can use this step to review and possibly revise them. If you do not yet have program-level student learning outcomes, you should plan on spending significant time on this step. You may wish to consult with appropriate offices at your institution (e.g. a Center for Teaching and Learning, a Director of Assessment, a Curriculum Committee, or Academic Affairs) about any institutional guidelines about program outcomes since these are often also used for institution-wide accreditation review.

Aspirational Program-Level Outcomes

Use your work from the prior steps to identify program level outcomes that would align well with your identity statement (1.3) and design principles (2.1, 2.2, and 2.3). What does your identity tell you about the characteristics you want to see in your graduates? What outcomes align well with your design principles. If you already have program level outcomes, you'll turn to those in the next step. Here, your goal is to reflect on what new or different outcomes you might set if you were designing a program from scratch based on your identity.

3.1. Based on your reflections, what program-level student learning outcomes would align with your identity and design principles?

Program-Level Outcome Revision

Now, if you have existing program level outcomes, compare those to your list in 3.1. Are there any existing outcomes you would like to remove, replace, or revise? Are there new outcomes you would like to add? If you do not have existing program level outcomes, use this opportunity to reflect on

what else might be missing. In both cases. iterate over your outcomes until you reach consensus within your department.

3.2. Give your new/revised set of program-level student learning outcomes.

Step Four: Shape of your Curriculum with respect to CS2023

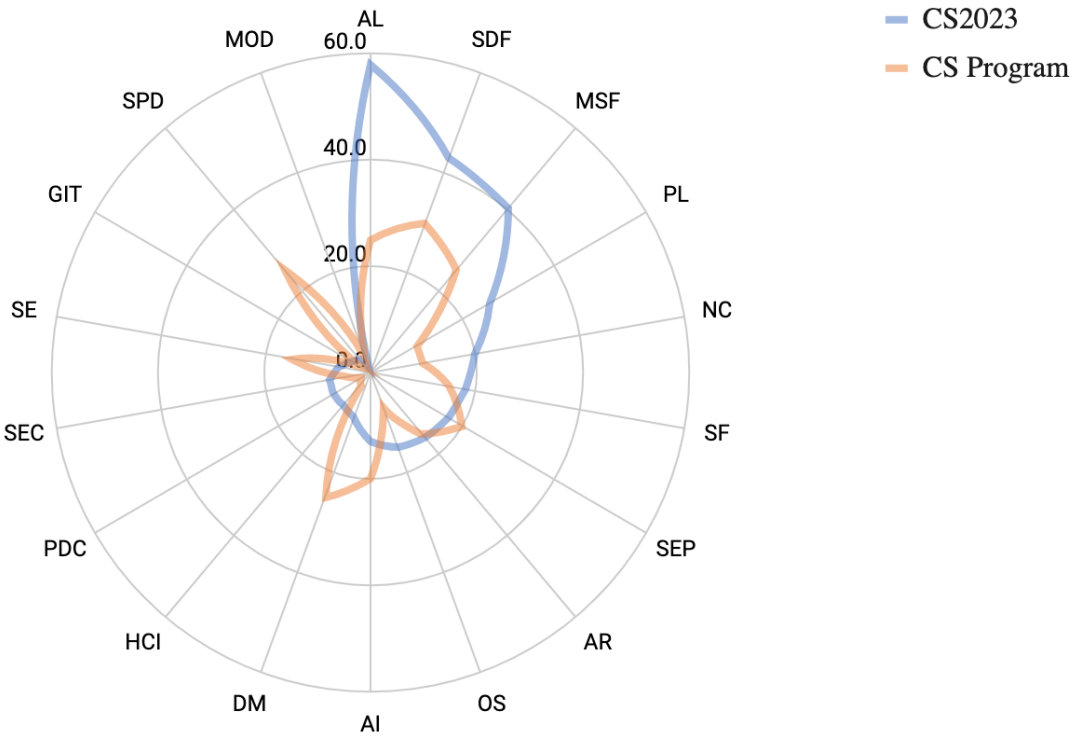
The CS2023 task force adopted two approaches to organizing their recommendations. One is a knowledge model that is organized around *Knowledge Areas* (KAs), for example: Software Development Fundamentals; Algorithms and Complexity; and Society, Ethics and Professionalism, to name just a few. The other is a competency model that focuses on the skills and professional dispositions that CS programs are expected to develop in their graduates. These might include perseverance, meticulousness, adaptability, etc. The goal in this section is to use metrics to develop a rough picture of how your current program aligns with the CS2023 recommendations. This rough picture can then be used to identify areas of difference. These differences can then be considered in light of your curricular design principles to identify areas for further investigation. Note that some differences may be well justified by your institutional and departmental contexts as formalized by your design principles. Others may represent areas where curricular revision should be considered.

The Knowledge Model

The drafts of the specific KAs planned for CS2023 are described at https://csed.acm.org/knowledge-areas/ . In the knowledge model, each of the KAs is further broken down into more specific *Knowledge Units* (KUs). For each KU, the recommended number of hours of coverage are given, called the *CS Core* hours. In addition, a number of *KA Core* hours are also recommended, which should be included in any program that offers depth in that KA. The documentation for each KA provides detailed lists and descriptions of the content that then makes up each KU. Those will be very helpful when considering revision and the design of specific courses, but that level of detail is not a part of the analysis in this step.

The CS Core

The radar diagram below shows the shape of the CS2023 CS Core and a hypothetical computer science program. Each radial dimension represents one of the KAs and the value in that dimension indicates the number of hours of coverage. The shape of the hypothetical CS program indicates that it has a heavier emphasis on some areas (e.g. SPD, SE, DM, and others) and a lighter emphasis on others (e.g. AL, SDF, MSF, and others).



4.1. Make a copy of the [CS2023-Workbook spreadsheet] to facilitate a review of your curriculum's current coverage of the CS Core KUs. *Note: The link to the spreadsheet has been removed for anonymous review, a screen shot of the spreadsheet appears in Appendix C.*

4.2. Open the *CSCore worksheet* in the CS2023-Workbook.

Within this tab, the *KA* and *Knowledge Area Name* columns identify each of the knowledge areas in CS2023. Each Knowledge Area Name links to the page for that KA in the CS2023 documents where more information about the KA can be found.

The *CS2023 Hours* column indicates the number of hours of instruction in the KA that CS2023 requires in the CS Core.

The remaining two columns in the CS2023-CSCore worksheet can be used to examine your program at two different levels of granularity depending upon your time and objectives.

- **Rough Picture:** Use the check boxes in the *Program Meets?* column to indicate each KA for which your program requires that all students receive at least the number of hours of instruction in that KA. Note that this will not update the radar diagram.
- **Detailed Picture:** Use the *Program Hours* column to enter estimates for the number of hours of instruction that all students are required to take for each KA. As you enter hour estimates the radar diagram will show the shape of your program as it compares to the CS2023 recommendations. Note: As you complete the CSCore worksheet, the CSCore CS2013 worksheet will show a similar radar diagram with both the CS2023 and the CS2013 recommendations displayed.

4.3. For which KAs are there differences (e.g. not meeting the required hours, or significantly exceeding the required hours) between your program and the CS2023 recommendations for the CS Core?

4.4 Optional Additional Depth: For each difference that you identified in 4.3, use the link in the CS Core worksheet to find the CS2023 document for that KA. That document will indicate the specific KUs that should receive coverage in the CS Core. How does your program's coverage of the KUs in this KA differ from the recommendations?

The KA Core

The CS2023 recommendations allow different programs to choose to emphasize different areas beyond the CS Core. For example, if your program offers a course, a topic within another course or a concentration in Software Engineering, then the *KA Core* for Software Engineering recommends specific KUs for additional hours of coverage beyond those in the CS Core.

4.5. Use the *KA Core* worksheet in your copy of the CS2023-Workbook spreadsheet to enter estimates of the hours of coverage for each KA beyond those in the CS Core. These hours may appear in either required or elective courses. What differences exist between your program and the CS2023 KA Core recommendations? *Note: The link to the spreadsheet has been removed for anonymous review, a screen shot of the spreadsheet appears in Appendix C.*

NOTE: The current draft of CS2023 provides only partial information about the KA Core.

Thus, the KA Core worksheet does not yet contain accurate values for all of the KAs. The KA Core worksheet will be updated as more complete information is released by the CS2023 task force.

The Competency Model

The current draft of CS2023 provides little information about the Competency Model. Thus, while we intend to include a section on it with an associated spreadsheet tool, there is insufficient information to do so at this time. This section will be updated as more complete drafts of CS2023 are released.

4.6. Appropriate questions will be added here as CS2023 drafts provide additional information about the competency model.

NOTE: The current draft of CS2023 does not provide details about the competency model. Thus, the Competency Core worksheet currently has no content. The Competency Core worksheet will be updated as more complete information is released by the CS2023 task force.

Analysis of Alignment with CS2023

The above questions have likely helped you identify ways in which your current program deviates from the targets set out in the CS Core, the KA Core (to the extent possible at this time), and the Competency Core (not yet possible, but will be added as CS2023 matures). In some cases your program may exceed recommendations and in others it may not reach them. Each of these deviations identify areas for consideration and reflection about whether those deviations are aligned with the mission, identity, situational factors and design principles that you identified earlier in this process. This should not necessarily be taken as a list of things that you must now add to your curriculum. Instead, you should reflect on where these differences in alignment reflect important elements of your identity or purposeful, strategic design choices based on your situational factors. For areas of significant deviation, it may be beneficial to do more detailed analysis of the hours allocated to KA and to the KUs within them. With that analysis you will be able to look both at where you may fall short on specific KU hours as well as places where you exceed target hours. This may help in finding opportunities to address shortcomings or allocate time to more fully address other priorities.

4.7. Consider the KAs in both the CS Core (4.3, 4.4) and the KA Core (4.5) and the Competency Core (4.6 when CS2023 matures) for which your program deviates from the recommendations. Explain how each of these deviations are or are not justified by your institutional and programmatic identities and your design principles?

4.8. Based on your analysis in 4.7, which deviations from CS2023 would make the most sense to address in a curricular revision? You should take into account the time available and the scale of revision you intend to undertake.

Step Five: Evaluating Your Current State

This step provides the opportunity for you to use existing data to inform your curriculum revision. At a minimum, you should identify areas of strength and areas for improvement with respect to your identity (Step 1), design principles (Step 2), programmatic outcomes (Step 3), and the CS2023 curricula (Step 4). You will also use existing departmental and institutional data to guide your revisions, or collect data that will help you identify strengths and weaknesses. Different programs may spend more or less time on this section depending on how much data is available and personal preference for focusing on the data. In particular, too many data streams may be overwhelming to those new to disciplinary-level curriculum development and revision.

Start by discussing how you are achieving the identity, design principles, and programmatic outcomes identified in steps 1-3:

- How does your current curriculum reflect the identity statement you described in 1.3?
- How does your current curriculum address the design principles you described in 2.1, 2.2, and 2.3?
- How does your current curriculum achieve the learning outcomes identified in 3.2?
- How does your current curriculum provide opportunities, or not, for addressing the priorities for curriculum revision you set in 4.7 and 4.8?

Next, identify and gather institutional and program assessment data. Examples of the types of data you may already have available are listed below. Consider consulting with Academic Affairs, Office for Institutional Research, Office of Institutional Effectiveness, or other similar groups to learn what specific data sources are available at your institution. You may also have an Institutional Fact Book that will indicate what data sources are available and provide summaries of the data.

1. Institution-Wide Data

    a. Standardized surveys such as the National Survey of Student Engagement (NSSE; https://nsse.indiana.edu/nsse/about-nsse/index.html) or Higher Education Data Sharing Consortium (HEDS) Surveys (https://www.hedsconsortium.org/heds-surveys/)
    b. Customized internal institutional assessment surveys,
    c. General education curriculum assessment data,
    d. Performance and retention (enrollment) data,
    e. Reports to and responses from Regional Accrediting Organizations
2. Departmental or Program Data
    a. External review documents (reports and responses)
    b. Current programmatic assessment goals, tools, and data
    c. Current curriculum map (a curriculum map associates student learning outcomes with specific courses; see https://www.colorado.edu/oda/sites/default/files/attached-files/curriculum_mapping_v2.pdf for information and examples)
    d. Departmental contributions to institution-wide assessment data (e.g.your department's assessment for institutional accreditation, enrollment data for your courses, etc.)
    e. Data for DIEA assessment, such as current and historical demographic data, disaggregated versions of these data sources, or specialized surveys/focus groups/etc.
    f. Annual reports to the Dean/College/etc.
    g. Surveys or focus groups with current students
    h. Feedback from graduates/alumni, including senior exit interviews, alumni "first destination" data, or alumni interviews
    i. Feedback from an industry advisory board or other advisory panel

If your program does not already have an assessment cycle, particularly a process that includes senior exit interviews and engagement with standardized surveys or institutional data, you may want to plan a separate time to examine and implement something like the National Center for Women and Information Technology's (NCWIT) Student-Evaluation of the Major: https://ncwit.org/resource/sem/ or their more general Evaluation tools: https://ncwit.org/resource/evaluation/

Review these data for strengths and weaknesses.

5.1. Based on your discussion of your identity, design principles, learning outcomes, and desired alignment with CS2023, combined with your data review, what are your program's current strengths?

5.2. Based on your discussion of your identity, design principles, learning outcomes, and desired alignment with CS2023, combined with your data review, what are places for improvement in your current program?

5.3 Based on your discussion of your identity, design principles, learning outcomes, and desired alignment with CS2023, combined with your data review, what program goals are currently not met by your curriculum?

Finally, set three to five specific goals for your curriculum revision. These should respond to the strength and weakness identified above, may incorporate high priority design principles, and help inform decisions. For example, McGill and colleagues[7] identified five goals for their Chemistry curriculum revision, including: “help students make connections between courses,” “articulate and incorporate scientific practices,” “maximize flexibility at the advanced level,” “increase student satisfaction with coursework” for both majors and non-majors, and earlier exposure to subdisciplines (p. 36).

5.4. What specific goals do you want to achieve in your curriculum revision?

Step Six: Implementation and Assessment

At this point, you should have four types of guiding documents prepared for your curriculum: your statement of program identity, your list of curriculum design principles, your program-level student learning outcomes, and your goals for your curriculum revision. You have also

[7] McGill, Tracy & Williams, Leah & Mulford, Douglas & Blakey, Simon & Harris, Robert & Kindt, James & Lynn, David & Marsteller, Patricia & McDonald, Frank & Powell, Nichole. (2018). Chemistry Unbound: Designing a New Four-Year Undergraduate Curriculum. Journal of Chemical Education. 96. 10.1021/acs.jchemed.8b00585.

considered how your curriculum aligns with CS2023 at a high-level and discussed what you would like to change and maintain about that alignment. Essentially, you have worked through the "data gathering" stage of a traditional curriculum review[8], with a focus on liberal arts identity and CS curriculum recommendations. We now combine implementation and assessment into one step - albeit a very large step. The process now aligns with traditional curriculum review[9]:

1. Develop course-level learning outcomes.
2. Map the curriculum: associate course-level learning outcomes with specific courses, and determine "routes" through the major, i.e., sequence the courses.
3. Plan assessment.
4. Action: plan and implement changes (e.g., create new courses, revise courses, etc).

Because this process is now well-documented elsewhere, we provide some general guidance in this section but also include references to other resources you may want to begin using.

We also note that at this stage, curriculum (re-)design could be applied at different scales; for example, do you want to focus on a specific knowledge area, a sequence of core courses, or target the full curriculum? In this step, we guide you through implementation and assessment by providing you with some guiding questions with connections to CS2023, and pointing you to existing resources. You should also consider reaching out to your Center for Teaching and Learning or other support you might have for curriculum and instructional design on campus. The documentation you have produced up to now will be a good starting point for a conversation with them about the resources they may have to help with things like developing curriculum maps.

Implementation of your curriculum (re-)design will necessitate iteration over Steps 1-5, and perhaps even a multi-year plan that sequentially addresses different program learning outcomes or course creation/revision. Consider creating a timeline for implementing curriculum changes. Be sure new and revised aspects of your curriculum always consider your identity (1.3), design principles (2.1, 2.2 & 2.3), and program outcome goals (3.2). Revisit the goals of your curriculum (5.4) (re-)design frequently.

6.1. What is your timeline for implementing your curriculum (re-)design? Your timeline could be based on curriculum review steps (above- course-level outcomes, map, assessment, action), program learning outcomes, course creation/revision, or other steps.

[8] Dyjur, P. & Kalu, F. (2018). Introduction to curriculum review. Taylor Institute for Teaching and Learning. Calgary: University of Calgary. https://taylorinstitute.ucalgary.ca/sites/default/files/Introduction%20to%20CR%20UPDATED%202019.pdf

[9] Ibid

Guiding Questions

Some high-level questions you will want to answer during your curriculum revision include:

6.2. What is the set of course-level student learning outcomes that you expect all of your majors to achieve? Where possible, organize or group these outcomes so that any sequencing of outcomes is reflected.

It is tempting to answer this question by thinking about the courses that make up your major and then articulating the learning outcomes for those courses. However, that risks reinforcing your current curricular structures and can make it harder to identify opportunities for innovation and improvement. While some or many of your courses may remain the same after a curriculum revision, you may also identify places where content can shift, different pathways through the majors can be created, or new combinations can be considered.

Start with your program-level outcomes and turn to CS2023 to consider what specific computer science skills or competencies are required to achieve that at the course level. Each Knowledge Area includes not just topics, but learning outcomes for that topic. Discussing these learning outcomes and considering which ones align most strongly with your program-level outcomes, as well as your programmatic identity, can provide helpful input to this process.

6.3. What is the organization of your course-level outcomes into courses? What courses exist (including in other departments or programs)? What courses will need to be modified? What courses need to be created?

6.4. What pathways will students take to complete these courses?

Resources

We encourage you to seek guidance from appropriate offices at your institution (e.g. a Center for Teaching and Learning, a Director of Assessment, a Curriculum Committee, or Academic

Affairs), At this step, we additionally recommend the following resources (also mentioned elsewhere in this document):

- Patti Dyjur, Kim Grant, and Frances Kalu, 2019, Introduction to Curriculum Review: https://docs.google.com/document/d/15LRmo2j8L-w21ZIVZTCAN22W795s4c-ZIUD72qfEz6o/edit#
- CU Boulder, 2019, Aligning Program Outcomes and Curricula through Curriculum Mapping: https://www.colorado.edu/oda/sites/default/files/attached-files/curriculum_mapping_v2.pdf
- L. Dee Fink, n.d., A Self-Directed Guide to Designing Courses for Significant Learning: https://www.bu.edu/sph/files/2014/03/www.deefinkandassociates.com_GuidetoCourseDesignAug05.pdf
- L. Dee Fink, 2003, Creating Significant Learning Experiences: An Integrated Approach to Designing College Courses (San Francisco: Jossey-Bass)

We have also included a set of questions to consider at the curriculum mapping step in Appendix B.

(Re-)Consider Assessment

Programs require an assessment plan to inform how well they are aligning with their identity, adhering to design principles, and achieving program outcome goals as well as for external evaluation and accreditation. A curriculum revision is not complete without a plan in place for assessment. Existing program assessment may require revision in light of your progress, and new programs will need a plan in place. Stakeholders to consider in your assessment include: introductory students, continuing students, graduating students, alumni, and faculty. Assessment methods include: standardized or customized surveys, focus groups, and standardized or customized skills assessment (such as ETS's Major Field Test: Computer Science). Consider what outcomes suggest success or opportunities for improvement. You will likely want to discuss your plans with your institution's assessment and research office.

6.5. How will you assess alignment with your identity? What outcomes suggest success and opportunities for improvement?

6.6. How will you assess adherence to design principles? What outcomes suggest success and opportunities for improvement?

6.4. How will you assess program learning outcomes? What outcomes suggest success and opportunities for improvement? Consider each learning outcome separately.

6.5. How will your assessment plan be used for external evaluation and accreditation?

Conclusion

This workbook has guided you through a process for designing or revising a liberal arts computer science curriculum that centers program identity and uses it as a lens through which curricular guidelines are applied. If you have completed this entire workbook, you and your colleagues will have reached a shared understanding of the following:

- Your unique programmatic identity as a liberal arts computer science program,
- Your identity-driven, structural, and DEIA-driven curricular design principles,
- Your programmatic student learning outcomes,
- Your level of desired alignment with the CS2023 curricular guidelines,
- Your current strengths as a program and most important places for improvement,
- Your plans for curricular change to achieve that improvement,
- Your plans for re-assessment and re-design in the future.

This process is an inherently iterative process, so we hope you will return to the steps in this workbook as you continue to review and revise your courses and curriculum. As noted in the introduction, depending on your goals, you can focus more on some steps than others. However, we do recommend that you always spend at least some time on the first three steps prior to jumping into steps four through six to ensure that you are proceeding with your curriculum work with a strong grounding in the identity, design principles, and curricular goals for your program.

The authors of this document welcome feedback on this process and suggestions for how to improve it. We would also welcome any examples of artifacts from this process that you would like to share with the broader liberal arts community. *Contact information will be shared in the de-anonymized version of this workbook.*

Appendix A: Design Principle Suggestions

Identity-based Design Principles

Design principles can also include institutional or departmental priorities for curricula, such as:

- Provide time for major exploration; allow students to begin the major as sophomores
- Connect to students' additional programs of study (double majors, minors, etc)
- Support internships
- Support study abroad
- Account for prior learning credit
- Provide service-learning or community work opportunities
- Provide paths for non-majors
- Prepare graduates for a wide variety of computing and computing-aligned careers (students may become developers, testers, documentation/training specialists, and or take on whole host of other technical and less technical roles based on their degree in CS)
- Support the needs of first generation undergraduate students
- Support the needs of transfer students
- Offer "quantitative reasoning" general education classes (separate from or same as major intro course)
- Offer general education classes besides quantitative reasoning
- Develop courses that contribute to other programs of studies on campus (majors, minors, list specifics)
- Utilize courses offered by other programs on campus
- Develop a "spiral" curriculum where students are introduced to topics in one course and revisit them frequently in other courses

They may also include some broad, content-specific design principles:

- Students develop skills in at least two programming languages
- Students participate in at least one larger-scale group software development project
- Students develop empathy for users, including understanding accessibility for different user abilities
- Students understand social, legal, and ethical concerns of technology
- Students understand their own role and the role of technology in a diverse, equitable, inclusive, and just society
- Prepare students for a particular career path (e.g. software development, cybersecurity, research)
- Incorporate experiential learning at the introductory level
- Incorporate supervised lab time into most core and elective courses

Structural Design Principles

Your institution may have rules in place for programs of study, such as:

- Size of the major (number of courses)
- Composition of the major (number of upper-level electives)
- Inclusion of a thesis or capstone
- Contributions to the general education curriculum (e.g., quantitative reasoning or writing across the curriculum)

You students may have particular characteristics you must take into account:

- Students tend to double major and seek additional credentials limiting the size of your program OR students tend to have a single concentration and you need to incorporate applications of computing.
- Students come in with variable experience in CS
- A large percentage of students come in on a pre-health or other pre-professional track
- Students have jobs/work study/athletics time constraints
- Students tend to work on their own OR students tends to collaborate/form study groups
- Students do/do not seek help in office hours
- Students have financial constraints that may limit the materials you can use
- Balance of residential and commuter students and ease of access to on-campus labs versus reliance on personal technology

You may have constraints based on the size (number of faculty or students) of your program or the University schedule that may affect the design of pre-requisite chains, paths through your program, variety of electives, or reliance on courses in other departments:

- Courses that can only be offered occasionally (e.g. every 3rd or 4th semester)
- Specific expertise of your faculty members
- Number of students needing elective options each year
- Student preferences regarding electives

You may have courses that are required for students in other majors are that have been committed to support college-wide requirements:

- List courses, requirements, and frequency of offering

In addition, consider physical space and personnel resources:

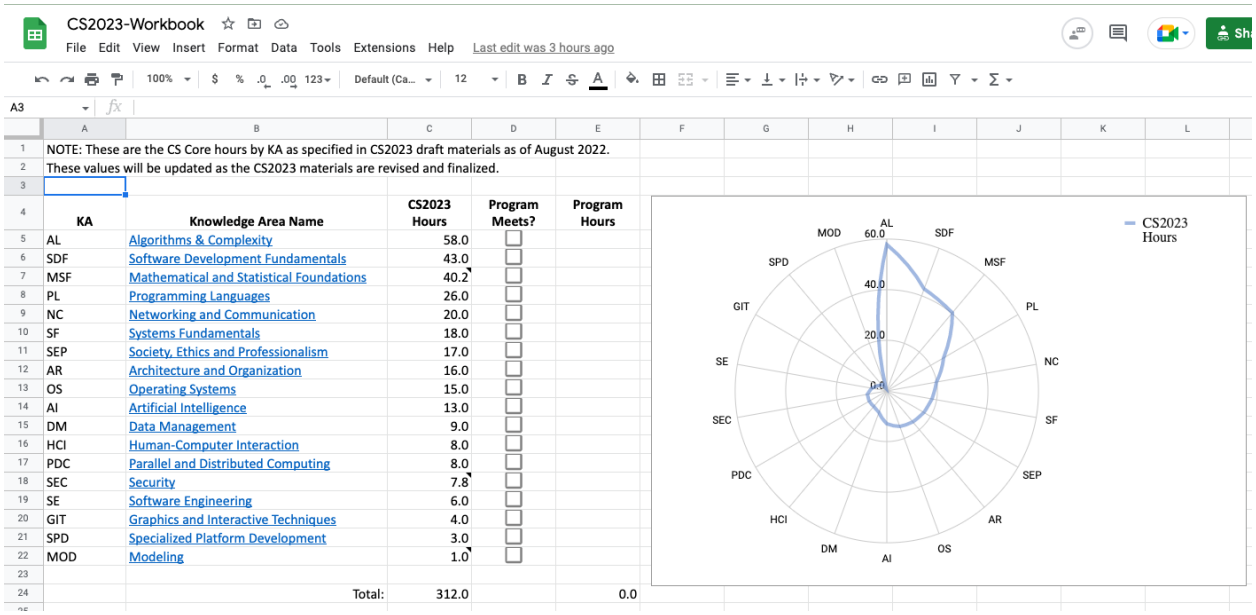
- Number of lab spaces and seats within labs
- Size and layout of classroom spaces
- Professional lab assistance or Information Technology support
- Student/peer teaching assistant support
- Office hours expectations

Appendix B: Suggestions for questions to consider during curriculum mapping (hint: refer to your design principles)

- Which course-level student learning outcomes are essential to cover early and how is that accomplished in your curriculum?
- How do you want to support students entering the major with a range of prior experience?
- Do you want all students to take a shared set of core courses, or do you want multiple options/pathways for achieving some of your course-level outcomes?
- How important is a shared introductory course versus multiple introductory options?
- How important is a shared capstone course or senior experience?
- Are any of your course-level outcomes covered by courses offered outside your program? (E.g. related to math, writing, communication, cultural competency, etc.)
- Which of your course-level outcomes complement each other well and could be effectively covered in the same course?
- What is the maximum length chain of prerequisites you want to permit for upper level courses in your program?
- Given your staffing, what is the maximum number of regular catalog courses you can reasonably offer?

Appendix C: The CS2023-Workbook Spreadsheet - Screen Shots for Anonymous Review.

The CS2023-Workbook Spreadsheet showing the CS Core worksheet.



CS2023-Workbook

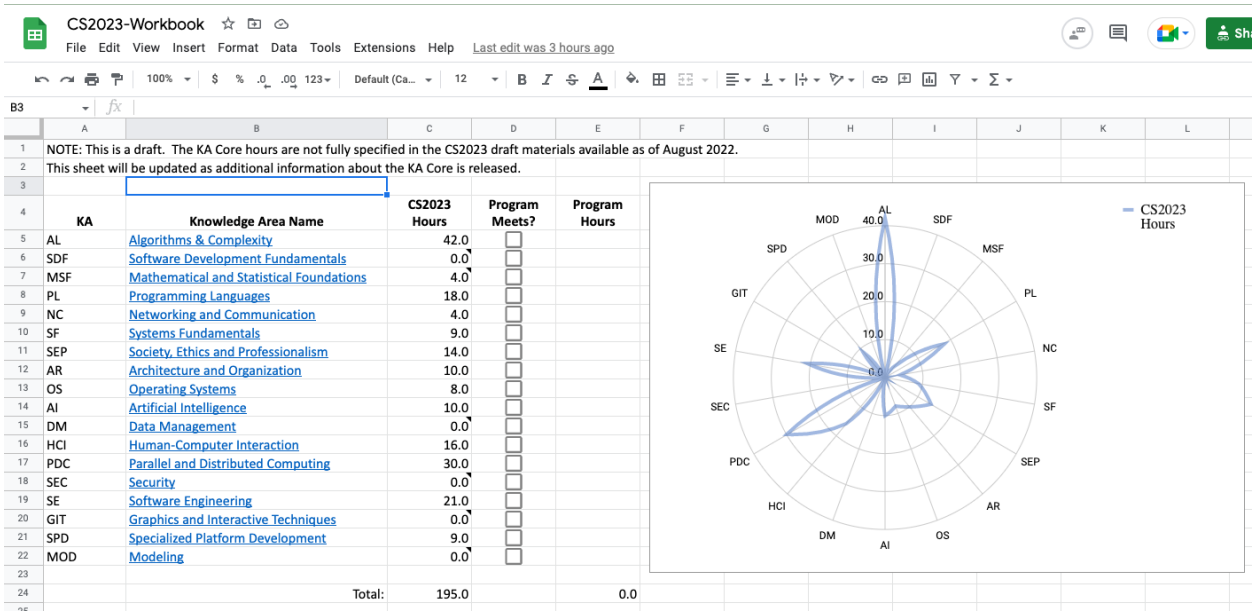
File Edit View Insert Format Data Tools Extensions Help Last edit was 3 hours ago

A3

NOTE: These are the CS Core hours by KA as specified in CS2023 draft materials as of August 2022.
These values will be updated as the CS2023 materials are revised and finalized.

KA	Knowledge Area Name	CS2023 Hours	Program Meets?	Program Hours
AL	Algorithms & Complexity	58.0	☐	
SDF	Software Development Fundamentals	43.0	☐	
MSF	Mathematical and Statistical Foundations	40.2	☐	
PL	Programming Languages	26.0	☐	
NC	Networking and Communication	20.0	☐	
SF	Systems Fundamentals	18.0	☐	
SEP	Society, Ethics and Professionalism	17.0	☐	
AR	Architecture and Organization	16.0	☐	
OS	Operating Systems	15.0	☐	
AI	Artificial Intelligence	13.0	☐	
DM	Data Management	9.0	☐	
HCI	Human-Computer Interaction	8.0	☐	
PDC	Parallel and Distributed Computing	8.0	☐	
SEC	Security	7.8	☐	
SE	Software Engineering	6.0	☐	
GIT	Graphics and Interactive Techniques	4.0	☐	
SPD	Specialized Platform Development	3.0	☐	
MOD	Modeling	1.0	☐	
	Total:	312.0		0.0

The CS2023-Workbook Spreadsheet showing the KA Core worksheet.



CS2023-Workbook

File Edit View Insert Format Data Tools Extensions Help Last edit was 3 hours ago

B3

NOTE: This is a draft. The KA Core hours are not fully specified in the CS2023 draft materials available as of August 2022.
This sheet will be updated as additional information about the KA Core is released.

KA	Knowledge Area Name	CS2023 Hours	Program Meets?	Program Hours
AL	Algorithms & Complexity	42.0	☐	
SDF	Software Development Fundamentals	0.0	☐	
MSF	Mathematical and Statistical Foundations	4.0	☐	
PL	Programming Languages	18.0	☐	
NC	Networking and Communication	4.0	☐	
SF	Systems Fundamentals	9.0	☐	
SEP	Society, Ethics and Professionalism	14.0	☐	
AR	Architecture and Organization	10.0	☐	
OS	Operating Systems	8.0	☐	
AI	Artificial Intelligence	10.0	☐	
DM	Data Management	0.0	☐	
HCI	Human-Computer Interaction	16.0	☐	
PDC	Parallel and Distributed Computing	30.0	☐	
SEC	Security	0.0	☐	
SE	Software Engineering	21.0	☐	
GIT	Graphics and Interactive Techniques	0.0	☐	
SPD	Specialized Platform Development	9.0	☐	
MOD	Modeling	0.0	☐	
	Total:	195.0		0.0