Computer Science Curriculum Guidelines: A New Liberal Arts Perspective

Amanda Holland-Minkley
ahollandminkley@washjeff.edu
Washington & Jefferson College
Washington, PA, USA

Jakob Barnard
Jakob.Barnard@uj.edu
University of Jamestown
Jamestown, ND, USA

Valerie Barr
vbarr@bard.edu
Bard College
Annandale-on-Hudson, NY, USA

Grant Braught
braught@dickinson.edu
Dickinson College
Carlisle, PA, USA

Janet Davis
davisj@whitman.edu
Whitman College
Walla Walla, WA, USA

David Reed
DaveReed@creighton.edu
Creighton University
Omaha, NE, USA

Karl Schmitt
Karl.Schmitt@trnty.edu
Trinity Christian College
Palos Heights, IL, USA

Andrea Tartaro
andrea.tartaro@furman.edu
Furman University
Greenville, SC, USA

James D. Teresco
jteresco@siena.edu
Siena College
Loudonville, NY, USA

ABSTRACT

ACM/IEEE curriculum guidelines for computer science, such as CS2013 or the forthcoming CS2023, provide well-researched and detailed guidance about the content and skills that make up an undergraduate computer science (CS) program. Liberal arts CS programs often struggle to apply these guidelines within their institutional context and goals. Historically, this has been addressed through the development of model CS curricula tailored for the liberal arts context. We take a different position: that no single model curriculum can apply across the wide range of liberal arts institutions. Instead, we argue that liberal arts CS educators need best practices for using guidelines such as CS2023 to inform curriculum design. These practices must acknowledge the opportunities and priorities of a liberal arts philosophy as well as a program's mission and identity. This paper reviews the context and motivation behind computing in the liberal arts. We also review the history of liberal arts CS educators and ACM/IEEE curriculum guidelines. We present data and trends about liberal arts computing programs, discussing how this informs curriculum design. Finally, we propose a process that guides programs to work with curriculum guidelines through the lens of institutional and program missions and identities, goals, and situational factors.

CCS CONCEPTS

• **Social and professional topics** → **Model curricula**; **Computing education programs**.

KEYWORDS

CS education, liberal arts, curriculum

SIGCSE 2023, March 15–18, 2023, Toronto, ON, Canada

ACM ISBN 978-1-4503-9431-4/23/03.
https://doi.org/10.1145/3545945.3569793

ACM Reference Format:
Amanda Holland-Minkley, Jakob Barnard, Valerie Barr, Grant Braught, Janet Davis, David Reed, Karl Schmitt, Andrea Tartaro, and James D. Teresco. 2023. Computer Science Curriculum Guidelines: A New Liberal Arts Perspective. In *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1 (SIGCSE 2023), March 15–18, 2023, Toronto, ON, Canada.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3545945.3569793

1 INTRODUCTION

In this paper, members of the SIGCSE Committee on Computing Education in Liberal Arts Colleges present an initial response to the new ACM/IEEE-CS/AAAI Computer Science Curricula currently under development (CS2023) [2]. Curriculum guidelines are designed to establish consistency among degree programs within a discipline. Yet curriculum guidelines also permit flexibility in that they do not necessarily specify the assignment of topics or learning outcomes to particular courses, as in a model curriculum [1]. Even with this flexibility liberal arts Computer Science (CS) programs often struggle to apply curriculum guidelines within their institutional context, mission, and goals. In particular, CS programs in liberal arts colleges must often balance adherence to curriculum guidelines with developing a unique curriculum reflecting the particular mission and identity of the program and institution. Historically, liberal arts CS programs responded to ACM and ACM-IEEE curriculum guidelines with their own model curricula (1986-2007) [7] or contributed curricular exemplars that demonstrated new flexibility within the guidelines (2013) [1]. But these responses do not capture the full range of computing programs in liberal arts colleges as characterized by Teresco and colleagues [17].

While liberal arts CS faculty have responded to past ACM/IEEE curriculum guidelines with model curricula for the liberal arts, **our position is that no model curriculum can capture the diversity of liberal arts computing programs.** Indeed, there is a risk that presenting one model curriculum or a small number of exemplars could guide programs towards needlessly restrictive views of what constitutes an appropriate CS curriculum. Moreover,

some small programs might dismiss even a liberal arts model CS curriculum as over-specified and inconsistent with their identity.

Instead, we propose a process for applying curricular guidelines, such as CS2023, to support many visions of collegiate computer science education. The process will support curriculum developers in striking a principled balance between enacting a program's particular vision and its alignment with curricular guidelines. A final version of this process and rationale is slated to appear in the companion volume on Curricular Practices to be published alongside the CS2023 curriculum guidelines [2]. A current draft is published on the Web[1] to enable feedback from the community.

This article presents our case as follows. We first discuss what sets liberal arts computing programs apart, both within their institutions and from computing programs at other types of institutions. We next discuss the resources liberal arts faculty have developed to respond to previous ACM/IEEE curricular guidelines. Then we use this context and history to argue for our position that the needs of the liberal arts computing community will be best met by a process for working directly with curricular guidelines rather than tailored models or exemplars. After presenting the results of a survey on the needs and priorities of liberal arts programs who are (re-)designing curriculum, we outline the proposed curriculum design process and preview its relationship to the forthcoming CS2023. Through this argument, we make the following contributions:

(1) a brief, updated history of liberal arts computing curricula;
(2) survey results showing liberal arts programs use curricular guidelines (e.g., CS2013) to inform their curriculum, but also integrate distinctive institutional and program identity such that they need flexibility in applying guidelines;
(3) introduction of a draft "Process Workbook" that will help programs work with curricular guidelines through the lens of institutional and program mission, identity, goals, and situational factors.

2 WHAT SETS LIBERAL ARTS COMPUTING PROGRAMS APART?

In the 2019 report of the SIGCSE Committee on Computing Education in Liberal Arts Colleges, Baldwin, Holland-Minkley, and Braught [4] synthesize many sources to characterize liberal arts colleges as "institutions that 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." Hence, a CS curriculum at a liberal arts institution must achieve two distinct but overlapping purposes: develop knowledge, skills, and professional competencies in preparation for further study or a career in computing *and* nurture a range of thinking skills applicable across many contexts.

Liberal arts CS programs are often not only set apart from other types of CS programs but also separated from other liberal arts programs. This separation is typically driven by the long-standing and recurring need to justify their existence and institutional belonging. Liberal arts CS faculty have had to argue that CS is not merely professional training but an academic discipline [9], and that CS "fit[s] within in the liberal arts" [18]. But the goals of CS education and liberal arts education are far from mutually exclusive [9]. A liberal arts context can improve the quality of a CS education, just as the liberal arts gain from including CS [18]. A liberal arts history major may not become a historian, just as a CS major may not become a computer scientist, but they both build a foundation for their future work and lives [18]. Indeed, advocates of "CS for All" efforts argue that "CS departments have, now, a unique opportunity to help smooth computing's transformation into a modern literacy" [22] that will support any educational goal. It is this focus on fundamentals and connections in liberal arts CS programs [18] that is a key element of a holistically developed student and a hallmark of liberal arts institutions.

At the same time, CS students benefit from the context of the liberal arts. Liberal arts CS students gain experience with collaboration and perspective-taking across disciplines, important for software development projects both in the academy and in industry [18]. Moreover, "breakthroughs in research often arise when a person connects different ideas in creative ways" [18]. Major curricula that ensure students study broadly outside of CS, as well as robust general education requirements, help students gain perspectives on social, legal, ethical, and diversity, equity, and inclusion (DEI) issues faced not only by the tech industry but by society more broadly. These benefits are well recognized through the demand for CS graduates with a liberal arts background. The media frequently features stories about the tech industry seeking out employees with liberal arts skills and alumni of liberal arts CS programs are regularly admitted into competitive PhD programs [14].

At some institutions, students are expected to commit to a major at matriculation. By contrast, liberal arts students are typically able or even expected to spend time exploring many possible majors, declaring a major by the end of the sophomore year. Liberal arts students often have significant educational goals beyond the completion of a major. These goals are supported by the construction of "flexible pathways," a distinguishing characteristic of liberal arts CS curricula [17]. Such pathways limit both the total number of courses required and the length of prerequisite chains to facilitate completion of a CS program within three years rather than four. They also allow minors, double majors, study abroad, or other supplemental curricular activities which support the liberal arts philosophy of education for all areas of life.

Some common features of liberal arts CS programs are often perceived as challenges or disadvantages compared to CS programs in large universities. For example, there is typically a small faculty, a limited range of course offerings, and fewer courses required for the major. A 2016 survey [4] found that liberal arts programs typically have a median of 3 faculty advising students in their program and graduate about 10 students per year. There is a roughly 50/50 split between programs offering a Bachelor of Arts (BA) versus a Bachelor of Science (BS) and very few programs are ABET accredited. A BA degree typically comprises about one-third of a student's total course requirements for graduation, with institutional limitations on how many courses can be required. These data are consistent with other studies of liberal arts curricula in the literature.

The view from within the liberal arts is that these perceived limitations are effects of a commitment to liberal education alongside institutional priorities and values, thus reflecting opportunities

[1] https://computing-in-the-liberal-arts.github.io/CS2023/

rather than limitations [17]. Liberal arts programs not only make space for student experiences outside of computer science, but value and prioritize those experiences. They recognize the ways in which robust general education requirements, off-campus studies, or second majors prepare students to be better computing professionals. Their curricula often focus on serving all students, not just CS majors. Thus, they work to incorporate external curriculum guidelines, but not to the detriment of their core educational values.

3 HOW HAVE LIBERAL ARTS FACULTY RESPONDED TO PAST CURRICULUM GUIDELINES?

Historically, ACM/IEEE curriculum guidelines have not adequately accounted for the distinctive characteristics of liberal arts CS programs. In response to this, a small group of faculty from liberal arts CS programs founded the Liberal Arts Computer Science (LACS) Consortium in 1984.[2] The first LACS model curriculum [11] was a response not only to the growth of interest in CS at selective liberal arts colleges, but also to dissatisfaction with prior ACM curriculum recommendations [7]. Early LACS members criticized ACM Curriculum '78 as "a collection of different programming and applications courses [that] fails to explicate adequately the principles that underlie the discipline" [11]. Moreover, the large number of courses required by Curriculum '78 was (and remains) incompatible with the structure of most liberal arts college curricula.[3] The 1986 "Model Curriculum for a Liberal Arts Degree in Computer Science" [11], developed by LACS members, sought to provide a principled basis for a model CS curriculum appropriate to the liberal arts college setting.

Over the following decades, LACS produced two further model curricula for liberal arts programs, in 1996 [20] and 2007 [8]. Each was a response to a preceding set of ACM/IEEE curriculum guidelines. In keeping with the philosophy of liberal education, LACS model curricula strove to emphasize "fundamental principles" as distinct from engineering practices [7]. Hence LACS model curricula were "not a pared-down version" of the ACM-IEEE model curricula, but rather "curricul[a] with a different emphasis" [7]. The programs implemented from these models were then tailored "to their environment and institutional goals rather than trying to replicate what non-liberal-arts programs do" [4].

CS2013 represented a significant departure from previous ACM-IEEE curriculum guidelines in three key ways.

(1) The CS2013 Steering Committee included three faculty from liberal arts colleges [4] [1], who were able to provide input on common practices and constraints at liberal arts colleges.[5]
(2) Where previous ACM-IEEE curriculum guidelines distinguished only between core and elective content, the CS2013 Body of Knowledge is divided into Core-Tier 1 and Core-Tier 2 topics, requiring coverage of only 80% of Core-Tier 2. This structure recognized that some computing curricula will include more or different required content than others [1].
(3) Rather than providing an abstract model curriculum, CS2013 included five "curricular exemplars." These exemplars are not models, but descriptions of actual major curricula offered by a range of institutions, including two liberal arts colleges (Grinnell College and Williams College).

Because liberal arts perspectives were well-represented in both the CS2013 development process and the resulting documents, LACS did not respond with a separate model curriculum. The curricular exemplars from Grinnell and Williams show that the recommendations of CS2013 are not only achievable by, but appropriate to, CS programs at some liberal arts colleges. With the caveat that Grinnell and Williams are highly selective liberal arts colleges with relatively large numbers of CS faculty [17]. Instead, LACS members who contributed to CS2013 and its exemplars responded with their experiences using CS2013 as a tool for curricular mapping and revision (e.g., [16, 19]) or with further example curricula that illustrate other approaches to implementing the guidelines (e.g., [15]).

In light of this history, CS2023 poses some new challenges for liberal arts computing programs. Unlike CS2013, CS2023 will comprise a "CS Core" and a "Knowledge Area (KA) Core." Programs are expected to cover all content in the CS Core and then to select Knowledge Areas for which they will cover the entirety of that KA Core as well. This gives less flexibility for courses and curricula to blend content from multiple KAs than is possible with CS2013's Tier 1/Tier 2 distinction. CS2023 will not include curricular exemplars; instead a companion volume will comprise contributed articles concerning implementation of the curriculum guidelines [2].

Moreover, the LACS Consortium as a voice for the liberal arts CS community has been superseded by the SIGCSE Committee on Computing Education in the Liberal Arts. This new voice is open to all ACM members who elect to join the Committee's mailing list.[6] The Committee is intended to provide representation and mutual support for liberal arts CS faculty [4]. Since its formation, members have met annually to discuss, among other things, their curricular "Opportunities, Challenges, Innovations" [6, 12, 13]. A recent synthesis of these materials [17] demonstrates that while the programs have many common constraints and goals, they are quite diverse in how they serve institutional missions and take advantage of local opportunities. CS2023 steering committee co-chairs approached leaders of the SIGCSE Committee on Computing Education in Liberal Arts Colleges to request a contribution to the CS2023 companion volume on curricular practices.[7] These steps towards constructing a more broadly encompassing liberal arts CS education community present an opportunity for a different kind of liberal arts response to CS2023.

4 WHY A PROCESS AND NOT ANOTHER MODEL CURRICULUM?

With the rapid changes in the field of computing, regular updates to CS curricular standards are needed and welcome. The history above also illustrates a positive trend in the ACM/IEEE curricular guidelines towards acknowledging the range of computing programs and anticipating the need for flexibility and customization. While CS2023 is still in draft form, the intent is to continue this trend by

[2] http://cs.wellesley.edu/~lacs/history.html
[3] David Reed, personal communication, July 22, 2022.
[4] Andrea Danyluk of Williams College, Richard LeBlanc of Seattle University, and David Reed of Creighton University
[5] David Reed, personal communication, July 22, 2022.
[6] See https://listserv.acm.org/
[7] Amanda Holland-Minkley, personal communication, March 3, 2022

delineating a 'core' of CS knowledge as distinct from elective areas. It will also describe multiple 'KA Cores', where programs can select particular Knowledge Areas to cover in depth. There are also plans for an increased focus on competencies and dispositions that has the potential to align well with the curricular approaches already taken by liberal arts institutions.

However, liberal arts CS programs can still perceive a mismatch between their own educational context and the context assumed by these curriculum guidelines. At the least, it is clearly the case that liberal arts institutions need to apply a significant level of interpretation when considering how to align their curriculum with these guidelines. Although recent guidelines state that it is not necessary to include a course in one's curriculum associated with each Knowledge Area [1], programs that are able and inclined to do so have an easier time finding course materials and demonstrating their alignment with the guidelines. The persistent inclination towards aligning courses with Knowledge Areas can clearly be seen in a recent paper that examined "coverage" of CS2013 Knowledge Areas in 500 colleges and universities [5]. Even though the paper quotes CS2013 as suggesting that KAs will be organized into courses "in different ways," and that KAs should not be equated to courses, the reported data collection process nevertheless matches course titles and descriptions to "representative course prototypes based on the eighteen knowledge areas" [5]. By contrast, liberal arts institutions can benefit from a process that supports curriculum designers in developing courses that blend KA content in unique ways.

Further, a review of current trends in liberal arts CS curricula [17] shows that programs apply and interpret curriculum guidelines in very different ways. These interpretations are driven by institutional context, including structural factors and consideration of the institutional mission. Proposing a single model for adapting the upcoming ACM/IEEE guidelines to a liberal arts context would act counter to this trend. It would falsely suggest there is a particular manner in which CS should be taught in a liberal arts setting.

At the same time, because of the difficult interpretive work they must do, liberal arts CS programs are in need of guidance on how to respond to, and use, curriculum guidelines such as CS2013 and CS2023. Accrediting bodies, external review processes, advisory groups, or others may require them to demonstrate alignment with national standards. Furthermore, like all other programs, they are concerned with remaining up-to-date and delivering effective, high-quality education. Curriculum guidelines such as CS2013 and CS2023 can be particularly beneficial to liberal arts programs in this regard because these programs often have smaller faculty and, thus, may not have in-house expertise on current trends across CS.

For these reasons, we recommend a departure from the practice of creating a model liberal arts CS curriculum, as in previous LACS responses, or promoting exemplary liberal arts CS curricula as in CS2013. Instead, we are developing a process that liberal arts CS programs can use to review and revise their curriculum in response to ACM/IEEE curriculum guidelines.

5 HOW CAN A PROCESS BEST SUPPORT LIBERAL ARTS PROGRAMS?

To inform development of a process to support liberal arts CS curricula we conducted a survey of liberal arts CS faculty. The survey assessed their experiences with curriculum design and revision, as well as their usage of and attitudes towards the ACM/IEEE CS2013 curriculum guidelines. Our survey was distributed through a call for participation sent to the SIGCSE Committee on Computing Education in Liberal Arts Colleges mailing list and Slack channel as well as the SIGCSE Members mailing list. Questions addressed structural characteristics of liberal arts CS programs, the usage of different sections of CS2013, and assessment of the relative importance and current coverage of the various KAs. Open-ended reflections asked their curricular philosophy and experience with curriculum (re-)design. Data collection occurred over three weeks in July 2022.

We received and analyzed 37 completed surveys. Multiple choice responses were summarized using descriptive statistics. A qualitative analysis was performed for open-ended questions. First major themes were identified, followed by a second review that annotated each response applying codes for all themes that appeared.

5.1 The current liberal arts landscape

Characteristics of liberal arts CS programs within the pool covered by our respondents are consistent with past findings. Programs remain small, with a median of 4 full-time faculty and 22 graduates per year. While these numbers are somewhat larger than in previous research [4], the relatively small number of students remains consistent with an emphasis on flexibility, customization, and personalized advising. Furthermore, the relatively small number of faculty continues to limit course offerings. Our sample includes slightly more programs offering a BA (69% of respondents) versus a BS (50% of respondents) degree. Similar to previous research [4], course requirements for BA programs account for about a third of the total courses required for graduation. Meanwhile, course requirements for BS programs account for up to half the course requirements. Thus, requirements for majors at liberal arts schools are small, typically no more than 10-11 courses at schools requiring approximately 32 courses for graduation.

Survey data also affirms the importance of viewing the CS curriculum as part of a larger institutional curriculum, and taking a broad view of the audience for CS courses. While 68% of respondents report offering an introductory course specifically for non-majors, 73% also report that non-majors take the same introductory courses as CS majors and minors. This suggests that introductory courses for majors and minors must often be designed to serve a diverse audience of students. Additionally, 67% report that their introductory courses also support their institution's general education requirements, and 57% offer courses beyond the introductory level that are designed to include non-majors. This reinforces that there is an interdisciplinary audience for individual CS courses and the CS curriculum as a whole at most liberal arts institutions.

When asked to describe the most unique or distinguishing characteristic of their current program or curriculum, the most commonly repeated theme was how pedagogy informed their curriculum (22% of responses). For example, programs referenced project-based or active-learning pedagogies as important features of their program. One program described the relationship between the distinctive pedagogical features of their program and their curriculum structure

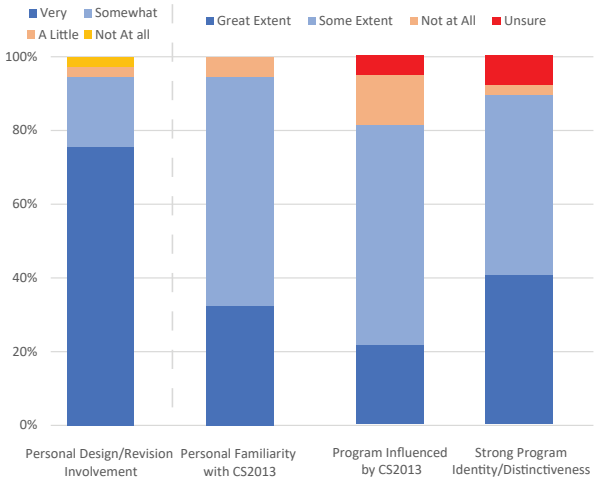

Figure 1: Liberal Arts programs are familiar with and influenced by curricular guidelines while striving for distinction.

as follows: "As a department, we're firmly committed to building inclusive spaces where we strongly message belonging. This includes frequent ethics discussions, interdisciplinary connections, active learning pedagogies, and replacing most tests with proposal-driven projects. The curriculum reflects that high-relationship approach: we have tried to keep the core minimal, and we rely on individual advising to steer students into the courses they need to pursue their interests." Other frequently occurring themes (at 16% each) were: the integration of social, ethical, and professional considerations within the curriculum; diversity initiatives; connections to courses outside of CS; flexibility within the curriculum; and having a strong focus on a selected special topic as a thread through the curriculum. These echo the themes from the literature about trends in liberal arts CS education, discussed in Section 2.

5.2 Needs of the community

We now turn to elements of the survey on how best to support liberal arts faculty in the process of reviewing their curriculum in the light of CS2023. Of the individuals surveyed, almost all (95%) had been involved in program-wide curriculum design and revisions, with 75% reporting that they had been very involved in this work. 94% were familiar with CS2013 and 81% reported CS2013 influencing their program to some or a great extent. These results are shown in Figure 1. We conclude that our sample has reasonable experience with CS curriculum design in a liberal arts context using ACM/IEEE curriculum guidelines.

Of respondents, 89% report their programs as having a distinct identity or mission to some extent or a great extent (versus not at all or unsure). The same percentage report that identity or mission plays a role in their curricular and course design decisions. This affirms our intention to center program identity and mission in the curriculum review process.

An important goal of our survey was to better understand how liberal arts programs used the CS2013 curriculum guidelines and what resources would be most helpful for future curriculum revisions. For each of the categories of content included in CS2013, respondents were asked to rate their usage as: not at all, to some extent, to a great extent, or unsure. Respondents report that they most frequently make use of the information about the content and learning outcomes in specific Knowledge Areas while making the least use of curricular exemplars for implementing the curricular guidelines. Even though CS2013 includes curricular exemplars from liberal arts institutions, qualitative data suggests these two exemplars are not enough. When asked what resources would be most helpful to the process of curriculum review and revision, participants most often (50% of responses) stated a desire for a broader range of example curricula from similar schools, supporting our contention that a small number of exemplars is insufficient. As one respondent said, "It would be great to have data on how our peer institutions have formulated their curricula. I don't like the idea of exemplars — these are just small points in a large data space that have somehow been called out for some reason that may not be relevant." Collectively, these findings support our strategy of providing a process for working flexibly with CS2023, rather than developing a single model liberal arts curriculum or curating a small set of exemplars.

The next most desired resources were guideline documents for CS curricula (40%) followed by peer advice (33%), which also suggests a desire to balance ACM/IEEE recommendations with liberal-arts informed guidance. Next, 30% of responses cited the value of assessment data, whether obtained through student/alumni/employer feedback or through formal external review. We similarly emphasize the importance of assessment and reflection in our process. Though less frequently mentioned, three respondents wrote specifically about the value of clearly understanding your situational factors as part of a revision and review process. This is an integral part of the process we describe.

Finally, we asked respondents to freely describe how they have made use of CS2013 in the design and revision of their current program and courses. The most frequent responses focused on reviewing the overall coverage within the program (54%) and making choices about what to emphasize and what to omit (43%). Furthermore, 32% referenced using CS2013 to review specific course content and outcomes, while 21% mentioned the small size of their program as a factor in how they used CS2013. Our process includes tools to help programs carry out a broad review of content coverage relative to CS2023, as well as reflective processes to support informed curricular decisions that reinforce an institution's and department's mission and identity.

6 A PROCESS FOR ALIGNING CURRICULA WITH CS2023

The Process Workbook under development is grounded in the curriculum design literature, particularly the key contributions of Fink [10], Wiggins and McTighe [21], and Appleton et al. [3]. It will make explicit use of CS2023 as a resource for curriculum assessment. It is designed to be usable by any program, regardless of size or staffing. It is scoped to support new curriculum design, the entire redesign of an existing curriculum, or focused revision of a portion of a curriculum. It explicitly accounts for the liberal arts philosophy and trends in liberal arts CS education by guiding programs to consider priorities they may have for interdisciplinarity,

socially-conscious computing, particular pedagogical approaches, or DEI in computing. Our ultimate goal is to guide curriculum authors within departments towards a curriculum (re-)design that addresses specific local factors and, more importantly, enhances the unique identity of the program.

Following Fink [10], the Process Workbook outlines an integrated design process adapted to the curricular level. The current draft of this work-in-progress comprises six major steps:

(1) articulate institutional and program mission and identity;
(2) develop curricular design principles driven by identity and structural factors, with attention to DEI;
(3) identify aspirational learning outcomes in response to principles and identity;
(4) determine the alignment of the current curriculum with CS2023;
(5) evaluate the current program, with attention to current strengths, unmet goals, and opportunities for improvement;
(6) design, implement, and assess changes to the curriculum.

In the first step, programs will reflect on institutional and departmental mission and culture, developing an identity statement that expresses their unique position within the landscape of liberal arts CS programs.

From there, programs will reflect on situational factors that must be accounted for in their curriculum design. This aligns with the results of Appleton et al's [3] review of theoretical considerations for curriculum development for technical education. Appleton et al highlighted the importance of a context-aware curriculum and the interrelationship between institutional culture and effective curriculum. Beyond structural factors such as staffing, programs are encouraged to address concerns about DEI and consider the role of their identity within their curriculum. This lets programs articulate the design principles that will be important in shaping their (redesigned) curriculum.

Next, programs will review or define program-level student learning outcomes. The draft Workbook guides programs to do this first without reference to CS2023, instead drawing on their prior reflective work, in accordance with the "Understanding by Design" and backward design processes of Wiggins and McTighe [21]. The goal at this step is to ensure a strong alignment between the program's identity statement and the outcomes they expect to achieve.

CS2023 will play a central role in the next phase when programs evaluate their curriculum as it currently exists and identifying goals for revision. The Workbook will provide tools to help programs investigate and visualize how their current curriculum aligns with CS2023. The tools will highlight where the curriculum completely covers the CS Core and selected Knowledge Areas, as well as gaps in coverage. Importantly, such gaps present an opportunity for reflection, but are not assumed to indicate a curricular deficiency. All of the prior work in the Workbook will be used as a lens for assessing whether it would support the program mission and identity to adjust the curriculum's alignment with CS2023 guidelines. Programs using our process should also expect to identify some areas where it is appropriate to deviate from CS2023. The process allows programs to document these as intentional design choices rather than accidental omissions.

Finally, the Workbook provides guidance for keeping these goals in mind, alongside the higher-order considerations of mission and identity, during the iterative implementation and assessment of curricular changes.

The draft Process Workbook is hosted on the authors' behalf by the Committee on Computing Education in Liberal Arts Colleges[8]. We are engaged in an iterative design process including several pilots at conference workshops and within individual departments that volunteer to test the process and provide feedback. The Process Workbook will continue to be refined based on this feedback and also as more details of CS2023 are released. A complete version will appear in the CS2023 Companion Volume on Curricular Practices[9]. In the meantime, we welcome feedback from the community.

7 DISCUSSION AND CONCLUSION

The Process Workbook we have described is a work-in-progress meant to illustrate our position on the role of CS2023 in curriculum (re-)design at liberal arts colleges. Responding to the community's desire for more examples rather than fewer (see Section 5.2), and building on previously developed repositories of liberal arts computer science curricula [6, 13], we also intend to share examples of how individual programs have used the process as a guide to others.

While our position has focused on the perspective of liberal arts colleges, this work can also support programs in other specialized educational contexts where curricular guidelines such as CS2013 and CS2023 potentially conflict with the context and priorities. We believe the state of CS education is improved if our curricula continue to evolve to reflect the diversity of institutions that provide CS education. As curriculum guidelines are most valuable when they are universally accepted, we urge the CS2023 Steering Committee to build adequate flexibility into the CS2023 guidelines so that widespread adherence is compatible with curricular diversity and innovation.

As stated in our position, no one model curriculum can capture the diversity of liberal arts CS curricula, but a process for applying curricular guidelines can support many visions of CS education. We hope this process will fill the gap between CS2013's call for creativity in how programs design their curricula and the need for principled guidance about how to undertake that task. In particular, this process will support programs in deciding what *not* to include and justifying those choices based on the foundational goals of the program and the institution. We foresee that this process will foster greater innovation within and variance among liberal arts CS curricula, which we believe strengthens computer science as a discipline.

ACKNOWLEDGMENTS

We thank the survey participants for their thoughtful responses. This work was done as part of ACM/IEEE-CS/AAAI Computer Science Curricula 2023 (csed.acm.org). We thank the CS2023 Steering Committee Co-Chairs for the opportunity to develop this work. We also thank Diane Boyd and Ben Haywood for their assistance with our research.

[8] https://computing-in-the-liberal-arts.github.io/CS2023/
[9] https://csed.acm.org/curricular-practices-volume/

REFERENCES

[1] ACM/IEEE-CS Joint Task Force on Computing Curricula. 2013. *Computer Science Curricula 2013.* Technical Report. ACM Press and IEEE Computer Society Press. https://doi.org/10.1145/2534860

[2] ACM/IEEE-CS Joint Task Force on Computing Curricula. 2021–2. CS2023: ACM/IEEE-CS/AAAI Computer Science Curricula. https://csed.acm.org/

[3] James Appleton, Frances Lawrenz, Elaine Craft, Wynn Cudmore, Jim Hall, and Jack Waintraub. 2007. Models for Curricular Materials Development: Combining Applied Development Processes with Theory. *Journal of Science Education and Technology* 16, 6 (2007), 491–499. https://www.jstor.org/stable/40188621 Publisher: Springer.

[4] 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 (April 2019), 22–29. https://doi.org/10.1145/3314027

[5] Richard Blumenthal. 2022. Alignment among Normative, Prescriptive, and Descriptive Models of Computer Science Curriculum: The Effect of ABET Accreditation on CS Education. *ACM Trans. Comput. Educ.* 22, 3, Article 35 (sep 2022), 27 pages. https://doi.org/10.1145/3513141

[6] Grant Braught, Douglas Baldwin, Janet Davis, Amanda Holland-Minkley, and Henry Walker. 2020. Liberal Arts Computing Curricula: Innovations, Challenges, and Opportunities, https://computing-in-the-liberal-arts.github.io/SIGCSE2020-PreSymposium-Event/. URL. SIGCSE 2020 Pre-Symposium Event by the SIGCSE Committee on Computing Education in Liberal Arts Colleges.

[7] Kim B. Bruce, Robert D. Cupper, and Robert L. Scot Drysdale. 2010. A History of the Liberal Arts Computer Science Consortium and Its Model Curricula. *ACM Trans. Comput. Educ.* 10, 1, Article 3 (March 2010), 12 pages. https://doi.org/10.1145/1731041.1731044

[8] Liberal Arts Computer Science Consortium. 2007. A 2007 Model Curriculum for a Liberal Arts Degree in Computer Science. *J. Educ. Resour. Comput.* 7, 2 (June 2007), 2–es. https://doi.org/10.1145/1240200.1240202

[9] R. N. D'heedene. 1982. Computer science in a liberal arts environment. *Proceedings of the Thirteenth SIGCSE Technical Symposium on Computer Science Education* 1, 1 (mar 1982), 232–236. https://doi.org/10.1145/800066.801374

[10] 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.

[11] Norman E. Gibbs and Allen B. Tucker. 1986. A Model Curriculum for a Liberal Arts Degree in Computer Science. *Commun. ACM* 29, 3 (March 1986), 202–210. https://doi.org/10.1145/5666.5667

[12] Amanda Holland-Minkley and Grant Braught. 2022. Innovations and Opportunities in Liberal Arts Computing Education, https://computing-in-the-liberal-arts.github.io/SIGCSE2022-Affiliated-Event/. URL. SIGCSE 2022 Affiliated Event by the SIGCSE Committee on Computing Education in Liberal Arts Colleges.

[13] Amanda Holland-Minkley, Janet Davis, Mario Nakazawa, Andrea Tartaro, and James Teresco. 2021. Innovations and Opportunities in Liberal Arts Computing Education, https://computing-in-the-liberal-arts.github.io/SIGCSE2021-PreSymposium-Event/. URL. SIGCSE 2021 Pre-Symposium Event by the SIGCSE Committee on Computing Education in Liberal Arts Colleges.

[14] Tom Perrault. 2016. Digital Companies Need More Liberal Arts Majors. URL. https://hbr.org/2016/01/digital-companies-need-more-liberal-arts-majors

[15] David Reed. 2017. Spiraling CS2013 Knowledge Units across a Small CS Curriculum. *J. Comput. Sci. Coll.* 32, 5 (may 2017), 125–131.

[16] David Reed, Andrea Danyluk, Elizabeth K. Hawthorne, Mehran Sahami, and Henry M. Walker. 2014. Experiences Mapping and Revising Curricula with CS2013. In *Proceedings of the 45th ACM Technical Symposium on Computer Science Education* (Atlanta, Georgia, USA) *(SIGCSE '14)*. Association for Computing Machinery, New York, NY, USA, 367–368. https://doi.org/10.1145/2538862.2538864

[17] 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 V. 1* (Providence, RI, USA) *(SIGCSE 2022)*. Association for Computing Machinery, New York, NY, USA, 537–543. https://doi.org/10.1145/3478431.3499329

[18] Henry Walker and Charles Kelemen. 2010. Computer Science and the Liberal Arts: A Philosophical Examination. *TOCE* 10 (01 2010).

[19] Henry M. Walker and Samuel A. Rebelsky. 2014. Using CS2013 for a Department's Curriculum Review: A Case Study. *J. Comput. Sci. Coll.* 29, 5 (may 2014), 138–144.

[20] Henry M. Walker and G. Michael Schneider. 1996. A Revised Model Curriculum for a Liberal Arts Degree in Computer Science. *Commun. ACM* 39, 12 (Dec. 1996), 85–95. https://doi.org/10.1145/240483.240502

[21] G. P. Wiggins and J. McTighe. 2005. *Understanding by design.* Association for Supervision & Curriculum Development, Arlington, VA, USA. http://books.google.com/books?hl=en&lr=&id=N2EfKlyUN4QC&oi=fnd&pg=PR6&dq=%22In+science,+they+use+their+senses+to%22+%22and+describe+the+characteristics+of%22+%22of+apples.+During+mathematics,+the%22+%22how+to+scale+up%22+an%22+%22to+make+a+quantity+sufficient+for+all+the%22+&ots=gn9Ej5SM3s&sig=4oN55BFQOECEmzSPkUKi9k1FIZQ

[22] Madeline Zug, Hanna Hoffman, Forest Kobayashi, Miles President, and Zachary Dodds. 2018. CS for All Academic Identities. *J. Comput. Sci. Coll.* 33, 4 (apr 2018), 130–137.