

Welcome to “Innovations and Opportunities in Liberal Arts Colleges”

Please sign in on the provided forms - walk-ins welcome!

Agenda:

1:00-2:00 Presentation Session A

2:00-2:15 Break

2:15-3:45 Presentation Session B

3:45-4:15 Break

4:15-6:00 Unconference Sessions

Event site with detailed agenda:

<https://bit.ly/4ao1ld6>



Innovations and Opportunities in Liberal Arts Computing Education

Sponsored by the
SIGCSE Committee on Computing Education in Liberal Arts Colleges

Supported by funding from NSF grant #2342587

A decorative light blue triangle is located in the bottom right corner of the slide, pointing towards the top right.

Presentation Session A (1:00-2:00)

Next up:

Short break (2:00-2:15)

Presentation Session B (2:15-3:45)

Break (3:45-4:15)

Unconference Sessions (4:15-6:00)



Welcome and Introduction

Report from the Essential Computing Concepts
Working Group

Latest Developments with Facilitated Curriculum
Design Support

Welcome and Introduction

SIGCSE Committee on Computing Education in Liberal Arts Colleges

<https://bit.ly/LiberalArtsCommittee>

SIGCSE-LIBARTS-COMM@listserv.acm.org

Our charge:

- Provide support and community for liberal arts computing educators
- Represent liberal arts computing to the broader CS community

Recent Focus

- Responding to CS2023 and supporting curriculum work
- Hosting events and working groups
- Identifying themes in liberal arts CS curricula
- Faculty recruiting and hiring
- Supporting student career development and opportunities
- Role of AI in CS education and of CS programs in AI education

Today's Agenda



Presentation Session A:

Welcome, Essential Computing Concepts Working Group, Facilitated Curriculum Design Support

Presentation Session B:

Graphics Education for Liberal Arts, Discussion-Based Assessment, AI as a GenEd Skill, Community Sharing, Breakout Discussions

Unconference Sessions:

Selection of Topics, Gathering of Working Groups, Debrief and Open Discussion
Proposed sessions document linked in the detailed agenda

Additional session proposals welcomed during the breaks

Report from the Essential Computing Concepts Working Group

Amanda Holland-Minkley, Washington & Jefferson College

Alyce Brady, Kalamazoo College

Megan Olsen, Loyola University Maryland

Latest Developments with Facilitated Curriculum Design Support

Jim Teresco, Siena University

Andrea Tartaro, Furman University

The Process Workbook

We have developed a curriculum design process involving CS2023 that:

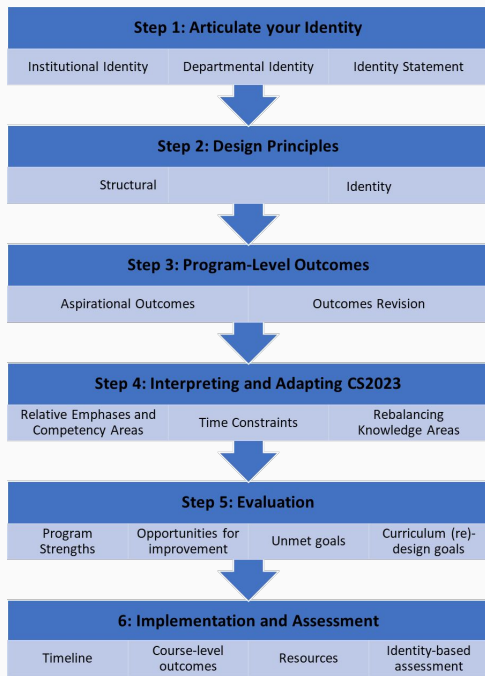
- encourages programs to reflect institutional and departmental identity
- leverages but is supportive of principled deviations from CS2023 curricular recommendations

A Workbook for Distinctive Computer Science Curricula: Designing Programs Aligned with Liberal Arts Institutional and Departmental Identity

Described in the CS 2023 Curricular Practice Volume

Computer Science Curriculum Guidelines: A New Liberal Arts Perspective

Process Workbook Steps



- Backwards-design approach for developing distinctive and institutionally-aligned curricula
- Early emphasis on institutional context, program identity, and curricular goals
- Step-by-step process presented through A *Workbook for Distinctive Computer Science Curricula*
- Explicitly incorporates CS2023 curricular guidelines through interactive spreadsheet tool
- All materials open source and available online

Using the Process Workbook

A department or program looking to update curricula could utilize the *Process Workbook* individually or take advantage of facilitation

- Workshops like today's unconference session or at other conferences (typically CCSC)
- Semester-long virtual facilitation through working groups
- In-person multi-day retreats

Step 1: Identity Statement

- “Our identity as a liberal arts computing program at W&J is to emphasize **computing practices in interdisciplinary contexts** and to view our **students as interdisciplinary learners** whose computing education will **complement and be complemented by their studies in other programs**. By prioritizing computing in context, we also emphasize a **focus on the end user/consumer** and the **individual/social impacts of technology**. We do not assume all students come to computing with the same values and priorities nor that they will all graduate with the same career and life goals.”
- **Institutional Vision** **Department Vision**

Step 2: Design Principles

Identity-Driven

- Major should have multiple entry-points that anticipate the range of interests and computing backgrounds of students.
- Program should offer courses of interest to students in other programs that can provide value as stand-alone courses or serve as entry points to the major or minor.
- Major should include applied and interdisciplinary projects at multiple points in the curriculum, including student-designed projects.

Structural

- Major must be able to be offered by 4-5 full-time faculty.
- Major must be reasonable to complete whether begun in the freshman or sophomore year.
- Major should enable faculty to occasionally teach in programs outside the department such as First Year Seminar, Gender & Women's Studies, etc.

Step 3: Program-Level Learning Outcomes

PLO #1: Research and analyze a technical challenge and make sound recommendations regarding its solution.

Aligned Identity Statement Concepts:

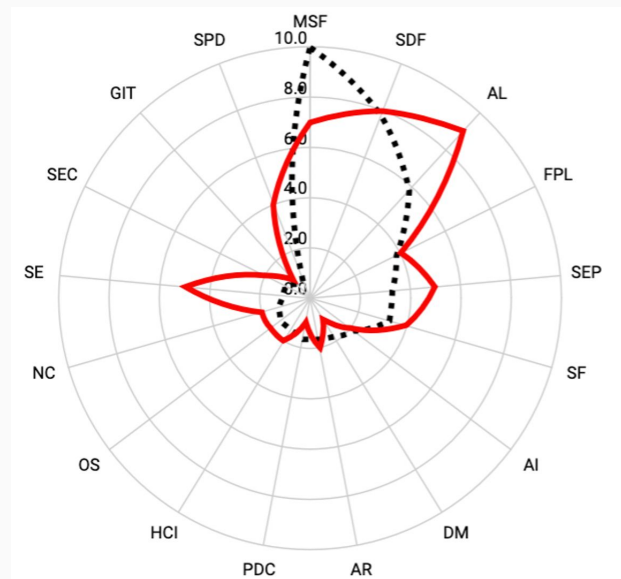
- Interdisciplinary computing
- Computing in context

Aligned Design Principles:

- Major should include applied and interdisciplinary projects at multiple points in the curriculum, including student-designed projects.
- Major should scaffold support for independent, creative problem-solving across the curriculum from the 100-level up to the 400-level.

Step 4: Interpreting and Using CS2023

KA	Knowledge Area Name	CS2023 Recommended Hours	CS2023		Relative Emphasis in My Curriculum										My Curriculum	
			Percent of CS Core	Relative Emphasis	Omit	Least								Most	Relative Emphasis	Percent of Curriculum
MSF	Mathematical and Statistical Foundations	55.0	20.37%	10.0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	7.0	11.48%
SDF	Software Development Fundamentals	43.0	15.93%	7.8	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	8.0	13.11%
AL	Algorithmic Foundations	32.0	11.85%	5.8	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	9.0	14.75%
FPL	Foundations of Programming Languages	21.0	7.78%	3.8	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	4.0	6.56%
SEP	Society, Ethics and the Profession	18.0	6.67%	3.3	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	5.0	8.20%
SF	Systems Fundamentals	18.0	6.67%	3.3	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	4.0	6.56%
AI	Artificial Intelligence	12.0	4.44%	2.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	2.0	3.28%
DM	Data Management	10.0	3.70%	1.8	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	1.0	1.64%
AR	Architecture and Organization	9.0	3.33%	1.6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	2.0	3.28%
PDC	Parallel and Distributed Computing	9.0	3.33%	1.6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	1.0	1.64%
HCI	Human-Computer Interaction	8.0	2.96%	1.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	2.0	3.28%
OS	Operating Systems	8.0	2.96%	1.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	2.0	3.28%
NC	Networking and Communication	7.0	2.59%	1.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	2.0	3.28%
SE	Software Engineering	6.0	2.22%	1.1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	5.0	8.20%
SEC	Security	6.0	2.22%	1.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	2.0	3.28%
GIT	Graphics and Interactive Techniques	4.0	1.48%	0.7	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	1.0	1.64%
SPD	Specialized Platform Development	4.0	1.48%	0.7	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	4.0	6.56%

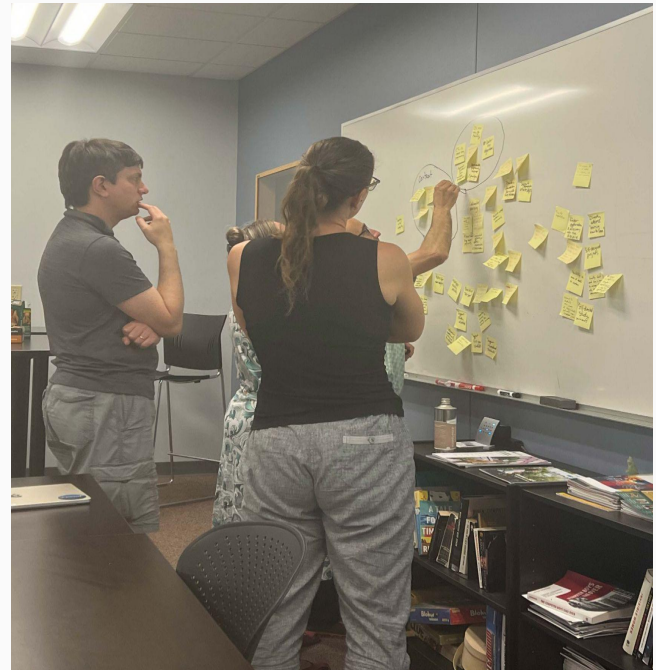


Pilot Working Group: Whitman College

Summer 2024

Multi-day facilitated on-site retreat

Whitman presented their curricular results at this affiliated event last year



Re-envisioning the Computer Science Major at Whitman College

William Bares, **Janet Davis**, John Stratton, Jordan Wirfs-Brock

Whitman College, Walla Walla, WA

February 26, 2025

Whitman CS major requirements, Fall 2017

The existing curriculum was designed to emphasize CS as a rigorous and distinct academic discipline and to be taught by only three full-time faculty on a 3-2 teaching load.

100	Introduction to Computational Problem Solving				
200	Computer Systems Fundamentals	Data Structures		Discrete Math & Functional Programming	Elective
300	Computer Systems Programming	Software Design	Algorithm Design & Analysis	Theory of Computation	
400	Senior Capstone Project				

Curricular Practices Workbook: Summer Workshop (4 days)

-1. Gather identity documents and self-assessment data; map existing curriculum

Monday	9-12	Introduction
		Ground rules/Community guidelines/decision-making process
	1:30-4:30	Step 1: Articulate identity
Tuesday	9-12	Step 2: State design principles
		Introduction to Step 3: Articulate program-level learning outcomes
	12-1(ish)	lunch together and informal discussion about program-level outcomes
	1:30-4pm	Step 3: Articulate program-level outcomes
Wednesday	9-11	Step 4: Interpret and adapt CS2023
	No afternoon meeting	Feel free to play around with the Step 4 curriculum tool some more.
Thursday	9-12	Step 5: Evaluate current state
	1:30-4pm	Step 6: Plan for implementation and assessment

Facilitated process

Each participant had a separate copy of the Workbook for individual work.

Our facilitator:

- Helped establish a timeframe for our activities
- Guided collaborative design activities
- Gave us assignments
- Guided our synthesis of individual work products
- Helped us recognize when collaborative products were "good enough"

Emergent goals for curricular revision

- Develop multiple entry points to the major
- Increase attention to social and ethical issues
- Make more room for electives (for students *and* faculty)
- Develop "spiral" elements of curriculum to build skills progressively
- Enhance opportunities for independent learning

Beyond the Workbook: Fall working group (1 hour weekly)

1. Envision new curricula as individual faculty
2. Identify common themes and courses
3. Map new and revised courses to CS 2023
4. Synthesize new major requirements,
aligned to program learning outcomes, vision, and design principles
5. Develop curricular transition plan
6. Propose new courses and major requirements
7. Revise learning outcomes (required by institutional process)

Learning outcomes

OLD - 2019

- Understand and apply fundamental algorithms and data structures;
- Understand the abstractions supporting modern software systems, and how the construction of those mechanisms affects the supported systems;
- Apply mathematical techniques to justify computational solutions and explore the limitations of computers;
- Communicate computational ideas through speech, writing, diagrams, and programs;
- Work with a team to design and implement a substantial, integrative project;
- Propose and compare multiple solutions to computational challenges, with consideration for the context and impact of each solution on the creators, maintainers, and users of that solution.

NEW - 2025

- Explain and apply algorithms and computational abstractions;
- Communicate effectively about computational concepts with diverse audiences in multiple modalities;
- Produce computational artifacts that reflect values of efficiency, validity, integrity, and beneficence;
- Demonstrate effectiveness and integrity in collaborating with diverse colleagues and stakeholders in team projects;
- Analyze and account for the implications of past, present, and future computational technologies on people, communities, and environments;
- Extend their knowledge and capabilities through self-directed learning.

BEFORE

100	Introduction to Computational Problem Solving				
200	Computer Systems Fundamentals	Data Structures		Discrete Math & Functional Programming	Elective
300	Computer Systems Programming	Software Design	Algorithm Design & Analysis	Theory of Computation	
400	Senior Capstone Project				

AFTER: Whitman CS major requirements, Fall 2025

100	Computer Systems & Society	Introduction to Computational Problem Solving	Mathematical Foundations of Computing (moved from 200- to 100-level)
200	Team Software Development	Data Structures & Algorithms	Elective(s)
300	Choice of: Human-Computer Interaction Computability, Complexity, and Algorithm Design Computer Systems Programming Software Performance Optimization		Elective(s)
400	Senior Capstone Project		

Challenges and limitations

- Significant time investment: ~60 hours/person over summer and fall
- External facilitation can be hard to obtain, but it is worthwhile
- New courses remain to be designed and implemented; tenure line to be filled
- ***Not a model curriculum, but a process exemplar***

Pilot Working Group: Kenyon

Kenyon College is working to design and implement a new

Interdisciplinary Program in Computing

The logo for Kenyon College, featuring the word "Kenyon" in a dark blue, serif typeface. The text is centered within a light blue rectangular background.

Pilot Working Group: Kenyon

A different use of the Process Workbook

- Not building a traditional Computer Science program
- Worked with the program director who taught a Design Thinking class to engage undergraduate students in the curricular design process
- Two site visits to engage with the class and other faculty stakeholders

Pilot Working Group: Kenyon

We want to have a program that focuses on interdisciplinary computing in a flexible manner that responds to the changes in computing and student needs.

1. Institutions: problem solving, analytical thinking, community.
2. dept: interdisciplinarity, transferable skills that will allow students to keep up with the rapidly evolving industry.
3. An ideal computing program in the Liberal Arts focuses on teaching skills with a longer shelf life. The chief of these are knowledge of systems and how technological systems affect people in everyday life.

Kenyon's computer science program encourages students to address ethical questions and make informed qualitative judgements, acquire quantitative skills and analyze data, and build strong collaborative skills ~~through~~ through an interdisciplinary approach to computing.

Pilot Working Group: Kenyon

Engagement with Kenyon demonstrates

- Utility of the Process Workbook outside of traditional CS programs
- Value of student input
- Value of an external facilitator
- Flexibility in the format of the facilitation

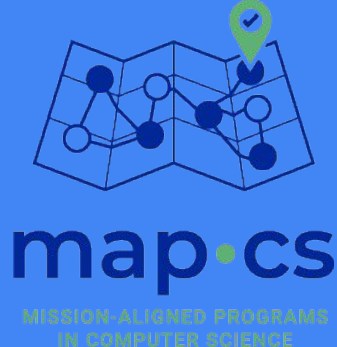
Funded Facilitated Support



map•cs

MISSION-ALIGNED PROGRAMS
IN COMPUTER SCIENCE

Funded Facilitated Support



- NSF-funded project to evaluate facilitated support for programs using the Process Workbook for curricular work
- Members of our research team will facilitate and observe the use of the Process Workbook
- Planning and recruiting underway
 - Semester-long virtual working groups
 - Multi-day retreat-format site visits



MAP-CS is supported by the U.S. National Science Foundation under Grant Nos. 2517967, 2517968, and 2517969.

Work with MAP-CS!

- Take a flyer
- Talk with any of the PIs: Amanda Holland-Minkley, Karl Schmitt, Jim Teresco, or other personnel: Andrea Tartaro, Jakob Barnard, Grant Braught
- Join our orientation workshop during the unconference later this afternoon
- Interest form: <https://tinyurl.com/map-cs-info>



Learn More About MAP-CS



<https://map-cs.org>

Presentation Session B (2:15-3:45)

Next up:

Break (3:45-4:15)

Unconference Session (4:15-6:00)



*Computer Graphics Wizard Academy:
Narrative-Driven and Shader-First Graphics
Education for Liberal Arts*

*Discussion-Based Assessment of Student Learning
in CS1*

Integrating AI Discernment in General Education

*Community Sharing: Open Positions and Other
Announcements*

Breakout Discussions

Computer Graphics Wizard Academy: Narrative-Driven and Shader-First Graphics Education for Liberal Arts

Sing Chun Lee, Bucknell University

Discussion-Based Assessment of Student Learning in CS 1

Adam Blank, Caltech

Zach Dodds, Harvey Mudd College

Julie Medero, Harvey Mudd College

Ben Wiedermann, Harvey Mudd College

Integrating AI Discernment in General Education

Mario Nakazawa, Berea College

Community Sharing

- Job openings?
- Relevant projects or resources?
- Opportunities?
- Additional topics of interest?

Breakout Discussions

Table A: Computer Graphics Wizard Academy

Table B: Discussion-Based Assessment

Table C: Facilitated Curriculum Design Support

Table D/E: Connect with Colleagues

Curricular Process Survey

Survey about institutional and departmental processes for curriculum revision and adoption

Sponsored by MAP-CS project to inform our work supporting the work of curriculum revision

Reviewed by Furman University IRB



<https://bit.ly/CurriculumProcesses>

Unconference Session

Selection of topics (4:15-4:30)

Session meetings (4:30-5:45)

Debrief and Discussion (5:45-6:00)

Proposed sessions descriptions linked
in the detailed event agenda:



A: Integrating AI Discernment in General Education

*B: Essential Computing Concepts: A Complementary
Structure to CS2023*

C: Implications of ACM Open for Smaller Colleges

D: MAP-CS: Process Workbook Orientation

*E: Hiring for Mission: Job Searches and Search
Committees in Teaching-Centered Liberal Arts
Computing*

Wrapping up.....

SIGCSE Committee on Computing Education in Liberal Arts Colleges

<https://bit.ly/LiberalArtsCommittee>

SIGCSE-LIBARTS-COMM@listserv.acm.org

Curriculum Process Survey

<https://bit.ly/CurriculumProcesses>

Thanks to NSF for funding support (grant #2342587)

