Assessment Report for the 2021-22 Academic Year

Department of Mathematics and Computer Science

Chair: Chris Camfield, Associate Professor of Mathematics

The department met on Thursday, May 19th for our year end assessment meeting. Present were Chris Camfield, Carol Ann Downes, Gabe Ferrer, Mark Goadrich, Lars Seme, and Brent Yorgey. Rebekah Aduddell was absent.

Part I: Program Assessment

In our Assessment Plan, 2021-22 is the year to examine Mathematics Learning Goal 2, Computer Science Learning Goal 2, and Computer Science Learning Goal 3.

We would like to point out that we collected data for every learning goal this year, not just those being assessed. In future years, we will have multiple years of data on hand when assessing those goals. The data is currently stored in a spreadsheet with a tab for each department learning goal.

MLG2: Understand basic content and principles in each of the broad divisions within mathematics: discrete (algebra and combinatorics), continuous (calculus and analysis), and geometric (linear algebra and topology).

Courses: All MATH courses except 115, 120, 215.

For direct assessment of this goal, instructors rated each student's performance in relation to this learning goal. Data has been collected for the past four semesters.

- A total of 371 students across 26 MATH courses were assessed using grades from relevant assignments and exams. The students were rated according to the following distribution:

Strong	Satisfactory	Needs Growth	Unsatisfactory	N/A
100	164	74	31	2

- The initial observation here is that 71% of students are performing at a satisfactory level or better. While this number is not terrible, we would like to see improvement.
- Since this learning goal involves three different areas of mathematics, we see a potential need to do a three-pronged assessment of this goal where each area is assessed within the appropriate courses. This will be part of a larger discussion of the learning goals in our external program review.

For indirect assessment of this goal, senior mathematics majors were asked the following questions in an exit interview. Seven students responded to the exit interview request, and the responses are tabulated below.

Question (5 = high, 1 = low)	**5**	**4**	**3**	**2**	**1**	**Avg**
How deeply did you explore discrete topics?	2	3	2	0	0	4.0
How deeply did you explore continuous topics?	3	2	2	0	0	4.1
How deeply did you explore geometric topics?	0	1	4	1	1	2.7
How clearly do you understand the distinction between them?	2	2	0	0	0	4.5
How clearly do you grasp how they integrate into a comprehensive view of mathematics?	3	3	0	0	0	4.5
How clear is your understanding of the motivation behind these aspects of mathematics?	1	1	5	0	0	3.4
How clear is your understanding of the aesthetics behind these aspects of mathematics?	2	3	1	1	0	3.9

- What stands out from these responses is that students felt like they properly explored discrete and continuous topics, but geometric topics were relatively neglected. This makes some sense since we have specific upper-level courses devoted to discrete and continuous mathematics, but not geometry.
- The mathematics faculty realize that we need to be more explicit in our upper-level courses about the geometric connections to the discrete and continuous mathematics we are focusing on. We do a fair amount of this in the 100 and 200 level courses, but we can draw more attention to it.
- We will also consider using our Advanced Topics course to introduce more geometry into our curriculum.

CSLG2: Use empirical methods to analyze computational systems and models.

Courses: CSCI 150, 151, 235, 270, 285, 320, 335, 340, 352, 370.

For direct assessment of this goal, instructors rated each student's performance in relation to this learning goal. Data has been collected for the last four semesters.

- A total of 373 students across 18 CSCI courses were assessed based on lab grades and assignments that involved analysis. The students were rated according to the following distribution:

Strong	**Satisfactory**	**Needs Growth**	**Unsatisfactory**	**N/A**
222	80	37	34	0

- The initial observation here is that 81% of students are performing at a satisfactory level or better.

For indirect assessment of this goal, a question was included in the course feedback survey. Data has been collected for the last four semesters.

- When asked about their perception of meeting this goal, 246 students across 17 CSCI courses responded according to the following distribution:

Strongly Agree	Agree	Neither Agree nor Disagree	Disagree	Strongly Disagree
137	81	14	4	10

- Students responded favorably at a rate of 89%, which is slightly higher than the faculty's opinion of their performance.
- The 10 students who responded with "Strongly Disagree" were all from fall semester sections of CSCI 150. This reflects some of the challenges related to an introductory course. We will investigate how we can better support students who are struggling in this class.

CSLG3: Employ multiple levels of algorithmic and data abstraction to manage system complexity.

Courses: CSCI 150, 151, 320, 322, 335, 340, 352, 360, 370, 382.

For direct assessment of this goal, instructors rated each student's performance in relation to this learning goal.

- A total of 328 students across 16 CSCI courses were assessed based on relevant projects and assignments with a substantial programming component. The students were rated according to the following distribution:

Strong	Satisfactory	Needs Growth	Unsatisfactory	N/A
175	66	38	48	1

- The initial observation here is that 73% of students are performing at a satisfactory level or better. Since this is a more challenging learning goal, it is not a surprise that these numbers are lower than other goals. There is room for growth as we investigate how we can better support students while working on large-scale projects.

For indirect assessment of this goal, a question was included in the course feedback survey. Data has been collected for the last four semesters.

- When asked about their perception of meeting this goal, 226 students across 16 CSCI courses responded according to the following distribution:

Strongly Agree	Agree	Neither Agree nor Disagree	Disagree	Strongly Disagree
138	70	8	2	8

- Students responded favorably at a rate of 92%, which is significantly higher than faculty opinion of their performance.
- The 8 students who responded with "Strongly Disagree" were again all from fall semester sections of CSCI 150. This further supports a need to investigate how we can better support students who are struggling in this class.

Part II: Actions Taken This Year

The following items were done this year in response to our to-do list from last year.

- In response to national trends, we have removed Calculus I as a requirement for the Computer Science major and minor. The course will still be accepted as an elective and is still a prerequisite for some electives that are more computational in nature, such as CSCI 285 *Scientific Computing.*
- We introduced new lab assignments in the Calculus sequence last year. While the general response to the labs was favorable, we made a few adjustments with student and instructor workload in mind. Instead of nine labs with written reports, we transitioned to three labs with written reports and six in-class engaged learning experiences.
- Detailed rubrics were developed for the direct assessment of six of the nine computer science learning goals. Those are attached as an appendix to this report.

Part III: To-Do List for 2022-23

- Align assessment plan and the way we conduct senior exit surveys in order to make better use of them and to make sure we are collecting the data we need about our program.
- For courses that are required for majors outside our department, inquire about what assessment is needed for those programs.
    - The following courses are required for majors outside our department:
        - MATH 130: Economics, Chemistry, Biochemistry/Molecular Biology, Physics, Chemical Physics
        - MATH 140: Chemistry, Physics, Chemical Physics
        - MATH 215: Politics
        - MATH 260: Physics
        - CSCI 150: Study of the Mind
    - The following courses are elective options for majors outside our department:
        - MATH 215: Biology, Health Science, Study of the Mind, Environmental Studies, Sociology/Anthropology
        - CSCI 151, 270, 285, 335: Study of the Mind
- Complete detailed rubrics for computer science learning goals CSLG 4, CSLG8, and CSLG 9.
- Look into rewriting mathematics learning goals using feedback from external program review.
- Consider adding Linear Algebra as an elective option for the Computer Science major.
- Evaluate options for the future of the mathematics capstone experience.

Appendix A: Rubric for Direct Assessment of Computer Science Learning Goals

Learning Goal	STR	SAT	NG	UNSAT	Assessment Tool
CSLG1: Create and demonstrate software that correctly solves realistic problems with open-ended scope.	Course project demonstrates an innovative solution to a challenging, realistic problem.	Course project demonstrates a solution to a realistic problem.	Course project is operational but partially incorrect in its solutions to problems.	Course project is largely incorrect in its attempt to solve a realistic problem, or the problem is not realistic.	Large course project(s)
CSLG2: Use empirical methods to analyze computational systems and models.	Analysis of solution convincingly shows correctness and/or time and space performance.	Analysis of solution shows correctness and/or time and space performance.	Analysis of solution mostly shows correctness and/or time and space performance but is significantly flawed in some manner.	Analysis of solution does not show correctness and/or time and space performance.	Course assignments with an analysis component
CSLG3: Employ multiple levels of algorithmic and data abstraction to manage system complexity.	Functions, classes, objects, and/or polymorphism manage project complexity in an innovative way.	Functions, classes, objects, and/or polymorphism manage project complexity in a competent way.	Functions, classes, objects, and/or polymorphism help manage project complexity, but are misused in some way.	Functions, classes, objects, and/or polymorphism fail to manage project complexity.	Complex course assignments involve use of functions, classes, objects, and/or polymorphism to manage complexity.
CSLG4: Employ mathematical ideas in a computing context.					
CSLG5: Create, implement, and evaluate software abstractions that model complex phenomena.	Complex phenomena are modeled in an innovative way.	Complex phenomena are modeled in a competent manner.	Modelling of complex phenomena captures some aspects but is inadequate in others.	Model fails to capture any essential elements of the modeled complex phenomenon.	Course assignments in which complex phenomena are modeled with data structures.
CSLG6: Create, apply, and understand the software abstractions that manage interactions with hardware.	Student code demonstrates a comprehensive understanding of the pertinent hardware.	Student code demonstrates a solid understanding of the pertinent hardware.	Student code demonstrates an understanding of some aspects of the hardware and a lack of understanding of other aspects.	Student code does not demonstrate any significant understanding of the pertinent hardware.	Course assignments in which student-authored code directly interacts with hardware.
CSLG7: As part of a team, develop robust software artifacts that successfully enable their users to achieve their goals.	Team develops software that enables users to achieve their goals, including implicit goals.	Team develops software that enables users to achieve their explicit goals.	Team develops software that enables users to achieve some but not all their explicit goals.	Team fails to develop software that enables users to achieve their explicit goals.	Large course project(s) conducted as part of a team
CSLG8: Employ written and oral communication in both technical and nontechnical settings.					
CSLG9: Understand the social and ethical context of computing.					