Student Outcomes Assessment and Success Report AY2020-21

Consult with your college dean's office regarding due date and how to submit. Deans will submit reports to the Office of Assessment & Accreditation annually by October 15.

Unit/Program Name: _____Computer Science MS_________ **Contact Name(s) and Email(s)** ____Jeff Kinne, jkinne@indstate.edu ____________

Part 1a: Summary of Student Learning Outcomes Assessment

NOTE: If data is missing due to COVID-19 transition issues, please describe these issues, their impact on your ability to assess student learning, and what, if anything, will change as a result.

Rating scale used for all assessed items: 3 Fully mastered, 2 Mostly mastered, 1 Somewhat mastered, 0 No or little mastery.
Covid impact on data collection – covid has impacted everything, including data collection. Due to higher workload overall, there has been a lower than normal completion of assessment tasks.
Additional notes – this is the first year of data collection using a revised list of learning outcomes, revised method of data collection, and revised major. It is likely there will be moderate revisions to all of these after discussing this preliminary data. We will also be discussing sequencing of topics in the revised major; the outcomes library will help with that discussion. For this collection cycle the focus was on outcomes shared between the CS BS and MS programs.

a. What learning outcomes did you assess this past year? If this is a graduate program, identify the Graduate Student Learning Outcome each outcome aligns with.	b. (1) What assignments or activities did you use to determine how well your students attained the outcome? (2) In what course or other required experience did the assessment occur?	c. What were your expectations for student performance?	d. What were the actual data/results?
1A - 1.B.i - Can program proficiently in the following programming paradigm - Imperative.	Asked on exit survey.	Average score between 2 and 3.	Average rating on exit survey of 2.5
1A - 1.B.iv - Can program proficiently in more than one programming paradigm. This should include at least two of the following. Event-driven.	CS579 - JavaScript assignment (3) – 9+ - Full mastery (3), 5-9 Mostly mastered(2), 1-5 – Some mastery(1), 0-1 No mastery(0).	Average score between 2 and 3.	Average rating in course of 3.
1A - 2.A.i Can use basic data structures (lists, stacks, queues, binary search trees, and hash tables) in writing programs.	Asked on exit survey.	Average score between 2 and 3.	Average rating on exit survey of 2.4.
1A - 2.A.ii Can use basic data structures (lists, stacks, queues, binary search trees, and hash tables) and answer basic questions on efficiency of these data structures.	Asked on exit survey.	Average score between 2 and 3.	Average rating on exit survey of 2.4.

1A - 2.B.ii - Can answer basic questions and reason about each of the following. Graphs.	CS558 - Use scores on some questions from the final exam, and will rate them as follows. 90-100% - fully mastered, 75-90% - mostly mastered, 30-75 - some mastery, 0-30 no mastery..	Average score between 2 and 3.	Average rating in course of 2.6.
1A - 3.A.i Can explain the basic internal workings of computer systems, including both hardware and software.	Asked on exit survey.	Average score between 2 and 3.	Average rating on exit survey of 1.75
1A - 3.A.iii - Can write correct assembly code for basic tasks.	CS556 - Third assignment: 9+ - Fully mastered(3), 5-9 – Mostly mastered(2), 1-5 – some mastery, 0-1 No mastery (0).	Average score between 2 and 3.	Average rating in course of 2.
1A - 3.A.iv - Can explain operating system and file system design concepts.	CS571 - Final, A – Fully mastered(3), B – Mostly mastered(2), C-D Some mastery(1), F – No mastery(0).	Average score between 2 and 3.	Average rating in course of 2.
1A - 3.B.ii - Can explain how operating systems share the following system resources among many processes, and tradeoffs between different approaches. Memory.	CS571 - Select questions from the Final, 9+ - Full mastery(3), 5-9 Mostly mastered(2), 1-5 Some mastery(1), 0-1 No mastery(0).	Average score between 2 and 3.	Average rating in course of 2.33.
1A - 3.B.iii - Can explain how operating systems share the following system resources among many processes, and tradeoffs between different approaches. Long-term storage.	CS571 - Maximum of assignment 7A/7B- 9+ Fully mastered(3), 5-9 Mostly mastered(2), 1-5 Some mastery(1), 0-1 No mastery(0).	Average score between 2 and 3.	Average rating in course of 2.33.
1A - 3.B.iv - Can write operating system code for a full-featured operating system.	CS571 - Maximum of assignments 6, 7A or 7B – 9+ Fully mastered(3), 5-9 Mostly mastered(2), 1-5 Some mastery(1), 0-1 No mastery(0).	Average score between 2 and 3.	Average rating in course of 2.33.
1A - 3.B.v - Can write code properly using system calls.	CS556 - First assignment, 9+ Fully mastered (3), 5-9 Mostly mastered (2), 1-5 some mastery(1), 0-1 No mastery (0).	Average score between 2 and 3.	Average rating in course of 1.5.
1A - 3.B.vi - Can write code for parsing network protocols and for implementing networked programs.	CS573 - Average of programming HW assignments. 85%+ fully mastered, 70%+ mostly mostered, 30%+ somewhat, < 40 none..	Average score between 2 and 3.	Average rating in course of 2.3.

1A - 3.B.vii - Can explain how data is packaged in various network protocols, and implications for reliability, security, and efficiency.	CS573 - Exam question. 90%+ fully mastered, 70%+ mostly, 30%+ somewhat, else none..	Average score between 2 and 3.	Average rating in course of 0.33.
1A - 3.B.viii - Can use program translation tools to build programming languages.	CS556 - 7th and 8th assignments combined – 15+ - Fully mastered (3), 10+ - Mostly mastered (2), 5+ - Some mastery (1) 0-4 No mastery (0).	Average score between 2 and 3.	Average rating in course of 2.5.
1A - 3.C.iii - Proficient at programming in specialized areas that are very common in industry, including the following areas. Networking.	CS573 - Average of programming HW assignments. 85%+ fully mastered, 70%+ mostly mostered, 30%+ somewhat, < 40 none..	Average score between 2 and 3.	Average rating in course of 2.5.
1A - 3.D.viii - Can apply the following standard algorithm techniques. Proficient at programming and algorithms analysis using these techniques. Dynamic programming.	CS558 - Programming assignment using DP algorithms, give rating of full mastery if > 9/12, mostly mastered > 7/12, some mastery > 4/12, no mastery if <= 4/12..	Average score between 2 and 3.	Average rating in course of 2.4.
1A - 3.D.x - Proficient at programming and using basic graph algorithms such as those for shortest path and minimum spanning trees.	CS558 - Use scores on some questions from the final exam, and will rate them as follows. 90-100% - fully mastered, 75-90% - mostly mastered, 30-75 - some mastery, 0-30 no mastery..	Average score between 2 and 3.	Average rating in course of 2.8.
2B.i - Can independently research a given computational problem to find the current state-of-the-art algorithms and implementations.	Asked on exit survey.	Average score between 2 and 3.	Average rating on exit survey of 2.14.
2C.i - Can work well in a group under the direction of a supervisor (faculty member or supervisor at work).	Asked on exit survey.	Average score between 2 and 3.	Average rating on exit survey of 1.9.

Note: *If you would like to report on more than three outcomes, place the cursor in the last cell on the right and hit "tab" to add a new row.*

Helpful Hints for Completing this Table

a. Use your outcomes library as a reference. Note any alignment with professional standards, as applicable.
b. Each outcome should be assessed by at least one direct measure (project, practica, exam, performance, etc.). If students are required to pass an examination to practice in the field, this exam should be included as one of the measures. At least one of the program's outcomes must use an indirect measure (exit interview, focus group, survey, etc.). Use your curriculum map to correlate outcomes to courses. Describe or attach any evaluation tools such as rubrics, scales, etc.

c. Identify the score or rating required to demonstrate proficiency (e.g., Students must attain a score of "3" to be deemed proficient; at least 80% of students in the program will attain this benchmark.)

d. Note what the aggregate level of proficiency actually was and the number of students included in the cohort or sample (e.g., 85% of the 25 students whose portfolios were reviewed met the established benchmark).

Part 1b: Review of Student Success Data & Activities

Use Blue Reports to generate the following information (as well as any other information helpful to you). A dashboard has been created in the Chairs view:

1) Cohort Sizes 2) Year-to-Year Retention 3) 5-Year Graduation Rate (undergraduate); Average time to completion (graduate)

Cohort sizes	*Computer Science MS (3060)*
Fall 2017	58
Fall 2018	34
Fall 2019	33
Fall 2020	34

The CS MS students have recently been around 80% composed of international students. Similar to other such programs, we saw a steep drop in cohort size at the beginning of the Trump administration and have also had to deal with reduced number of applicants during the covid pandemic. Keeping the numbers that we do have has been a result of a number of efforts – putting the program online (as well as face to face), allowing international students from certain countries to start online during the pandemic (based on a one time exception from the government), and creating new concentrations in data science and bioinformatics.

Retention % (Next Fall)	**Fall 2017**	**Fall 2018**	**Fall 2019**	**Fall 2020**
Computer Science (3060)	87.50%	80.00%	95.65%	100.00%

Retention in the CS MS program has traditionally been very high, close to 100%. We have not looked into the data of what happened with the Fall 2017 and Fall 2018 classes.

Average Years to Graduation	**2017-2018**	**2018-2019**	**2019-2020**	**2020-2021**
Computer Science (3060)	1.9	1.8	1.9	1.8

The CS MS has enough flexibility that students normally do not have trouble graduating in 2 years full time, and some finish in 1.5 years. Most students are full time, so the longer time to graduation for part time students does not impact the "years to graduation" number much.

What worked well in supporting student success this year?

The need to deliver content online due to covid has had both negative and positive impacts, on both students and faculty. On the positive side, since we now know how to do courses online we have decided to make our programs available via distance (which gives additional flexibility to campus students as well). On the negative side, student performance in early courses seemed to be even more bimodal than normal in the past year.

What are the most significant opportunities for improvement upon which to focus in the coming year?

For the spring and fall 2021 we have had all entering students take the same set of courses their first term – regardless of their intended concentration, and regardless of whether they think they need the beginning courses we are putting them into. We had been seeing too many students choose the wrong set of courses based on an incorrect self-evaluation by the students. We will do this again for the spring and evaluate how the fall and spring 2021 students are doing, and may adjust which set of courses we have them take. We may also develop a "test out" process to determine which students need which courses.

Part 2: Continuous Quality Improvement

Reflect on the information shared above regarding student learning, success, and career readiness. In no more than one page, summarize:

1) ***the discoveries assessment and data review have enabled you to make about student learning, success, and career readiness*** *(ex: What specifically do students know and do well—and less well? What evidence can you provide that learning is improving? How might learning, success, and career readiness overlap? What questions do your findings raise?)*
2) ***findings-based plans and actions intended to improve student learning and/or success (expansion of Part 1a, box e as needed)***
3) ***what your assessment plan will focus on in the coming year***
4) ***how this information will be shared with other stakeholders***

We are really at the beginning of the assessment cycle for the computer science MS. The major was revised to add a new concentration (data science). As mentioned above, we have changed the intake process for new students and which courses they are allowed to take their first term. The outcomes library was updated to have more specific outcomes that should be more easily directly assessed. The assessment plan was changed so that assessment data would be collected in courses throughout the major by most of the faculty. Once we have this working well we should have better quality data that is more specific and at multiple points of their studies.

As of fall 2021, we are at the stage of collecting data for one year and having some initial feedback from faculty on the whole process. The following are some initial suggestions from faculty after having gone through the process once.

1. Changes to which outcomes are contained in which courses. We are currently having all new students take CS 500 Programming Fundamentals (C programming and data structures) and CS 501 Programming for Data Science (python programming), and CS 600 Concrete Mathematics if their first fall term. Some discussion has begun about the balance between programming and algorithms/theory in CS 500. This will be impacted by any "test out" process that we develop.
2. Changes to the outcomes themselves. Some of the outcomes are not worded in a way that lines up well with what is happening in the courses. The outcomes were written down mostly by one faculty member, so having some changes based on what is happening in the courses makes sense.
3. Reminders at the right time of the semester to collect data.
4. Organize outcomes and data to be more useful. In particular, put all of this into a database to make it easier to answer questions about which students have mastered which skills.

5. Define different levels of “acceptable” for students finishing the program. In particular, some students will have good skills in one particular area (e.g., web programming) but not meeting expectations in other areas (e.g., computer systems or algorithms), which may be sufficient for getting the job they want.
6. Differentiate (or not) “acceptable” levels for students who are not majoring in computer science, in particular for courses with high interest from outside of CS.