

Computer Science Assessment Report 2018

0.1 Introduction

The computer science department was rebooted last year. All the courses are in the process of being rebuilt from the ground up. Our assessment plan was brand new last year as well. It was rough and imperfect.

This new iteration of the plan is more refined and focused. In fact, the plan this year is sufficiently improved from last year's that we've decided to structure this year's report on the new assessment plan instead of the 2017 version. The data for this report is incomplete, since the plan was not finalized until late in the year. However, the available data still paints a useful picture. We look forward to next year's report when we can collect more data and start to look for trends compared to this year's data.

0.2 Program Assessment Reort

In this section we look at which assessment methods for each learning goal achieved their success criteria and which did not. These results will be presented in the identical format the assessment plan. It is recommended to view both documents at once for the best readability.

We provide some qualitative context where data is lacking or incomplete. Under the assessment method headings we will describe what data we collected. Under the success criteria headings we will discuss whether the prescribed success criteria were met and what we plan to do to improve.

I. Goal 1: Technical Expertise

This year, we focused on CS132 Advanced Design to track this learning goal.

A. Assessment Method 1: Testing in CS131, CS132, CS133, CS134

We didn't track the exam data as our plan mandates, but we can still state some useful numbers. See below.

1. Success Criteria 1: Exam Grades

Out of a total of eight students, three exhibited excellent technical expertise on exams, three were mediocre, and two were bad. One of the mediocre students and one of the bad ones were at least partially the result of poor effort. Still, overall this is a bit below where we want to be. This can likely be improved by some better lesson planning next year.

B. Assessment Method 2: Projects There were two separate student projects in 132 this year.

1. **Success Criteria 1: Project Grades** Due to the overhaul of our assessment plan, the projects this year were not graded according to the required rubrics described in the plan. However, the breakdown of qualitative performance was identical to that of the exam performance: three good, three mediocre and two bad. Again, this can likely be improved by some better lesson planning next year.

C. **Assessment Method 3: Performance on External Coding Apps** The students in CS23 Object Oriented Programming were introduced to CodeWars late in the spring semester.

1. **Success Criteria 1: Tracking Progress** It was too late to actually track progress, but a handful of students started using the apps eagerly. This is a good sign for our prospects of future engagement.

II. **Goal 2: Design Expertise** We used CS132 for this learning goal as well.

A. **Assessment Method 1: Projects** As mentioned above, there were two projects assigned in CS132.

1. **Success Criteria 1: Project Grades** Again, the projects this year were not graded according to the required rubrics described in the plan. Qualitatively, out of the eight students, there were four good, three mediocre, and only one bad. Design expertise didn't seem to be as much of a challenge for this group as technical expertise.

III. **Goal 3: Theoretical Knowledge**

A. **Assessment Method 1: Grades in Discrete Mathematics, Algorithms, and Linear Algebra.** We used Discrete Mathematics for this assessment method.

1. **Success Criteria 1** The average grade in the course was 2.9, or barely below a B. According to this success criteria as described our assessment plan, this is a failure. We might have to re-think this success criteria because some of the lower achieving students weren't computer science students, which skewed the outcome.

A. **Assessment Method 2 Qualitative Professorial Observation** We used CS233 object oriented programming for this one.

1. **Success Criteria 1:** Theoretical knowledge was pretty good in this class. As a concrete example, I would say that all but two or three out of twelve students had recursion more or less at their fingertips.

IV. **Goal 4: Student Buy-In**

A. Assessment Method 1: CS131 Engagement

1. **Success Criteria 1: CS131 Enrollment** We had nearly 20 students this year, which is a good sign.
2. **Success Criteria 2: Major Declaration Rate** CS132 enrollment was lower than we'd like at eight. Also, our of those eight, maybe four have what it takes to major in CS. We'd like those numbers to be higher. Our sophomore majors look good for now at 5 or 6. Recall that this is the first class to enter into the new major, we'll expect these numbers to grow in the next 2-3 years.

A. Assessment Method 2: Major/Minor Retention

1. **Success Criteria 1: Major/Minor head count** This number won't be meaningful until this year's rising juniors graduate. This is because the current juniors and seniors came up under the previous CS program, so their numbers don't reflect our current efforts.

V. Goal 5: Professional Success

A. Assessment Method 1: Proportion of Graduating Majors Continuing in Computer Science This number won't be meaningful until this year's rising juniors graduate, for the same reason detailed above. However, the one graduating senior who asked me for a recommendation did get the programming job he wanted. This is a good sign.

1. **Success Criteria 1:** N/A

A. Assessment Method 2: Internships Our summer internship advising has not yet been implemented. It should be in place by the end of next year, at which point we can begin to track data.

1. **Success Criteria 1:** N/A