Computer Science Assessment Plan

0.1 Introduction

The computer science assessment plan consists of a mission statement and a list of learning goals. The learning goals are measurable features of our computer science students that can be used to assess student learning in a meaningful way.

Each learning goal has one or more assessment methods. The assessment methods are measurements we use to determine whether the learning goals are being met.

Each assessment method has one or more success criteria. The success criteria specifies what results from the assessment methods are considered satisfactory.

0.2 Computer Science Mission Statement

Our mission is to produce students with both theoretical and technical expertise in the field of Computer Science. Our students will learn how to formulate a goal, come up with a design plan to implement the goal, and then execute the plan successfully. They will also gain an appreciation for the abstract mathematical beauty of Computer Science.

0.3 Computer Science Learning Goals

In this section we identify and define our learning goals.

Technical Expertise

The term "Technical Expertise" refers to the mechanical and syntactic aspects of computer science. For example, successfully writing a recursive function to accomplish a specific task is an instance of technical expertise, while understanding the theory behind why recursion works is not.

Theoretical Knowledge

This learning goal refers to the mathematical theory behind Computer Science. Understanding why recursion works is an instance of theoretical knowledge.

Design Expertise

Design Expertise can be thought of as the midpoint between technical expertise and theoretical knowledge. Design expertise is the ability to start with an abstract goal and turn it into a specific, practical strategy that can be implemented within the framework of some technological platform.

Student Buy-in

We believe that if our computer science students are learning at a high level, then the major will be poplar and the students within the major will display enthusiasm for their work. It follows that tracking student morale and engagement is an indirect measure of overall student learning.

Professional Success

If our learning goals are being met, our students should be finding success after graduation. If our students are not finding success after graduation then something is certainly wrong whether or not our other stated learning goals are being met.

0.4 Program Assessment Plan

In this section we lay out our learning goals, assessment methods, and success criteria in an outline format.

I. **Goal 1: Technical Expertise**
Students will develop proficiency in the "nuts and bolts" skills of various programming languages and technologies. They will be able to recognize which technique is appropriate for a given task and then implement that technique correctly using best practices.

   A. **Assessment Method 1: Testing in CS131, CS132, CS133, CS134**
   Every class will have multiple midterm exams as well as one final exam. These exams will require students to solve problems individually with no collaboration. Incorrect implementation of solutions will result in lower scores.

      1. **Success Criteria 1: Exam Grades**
      We will track students exam grades. Ideally the class average on an exam will fall between 80-90. We recognize that there are other contributors to exam grades than technical expertise: a student might lose points due to "big picture" misunderstandings. So an exam grade below 80 does not necessarily indicate a failure of technical expertise. However, exam grades above 80 do indicate a satisfactory level of technical expertise.

B. **Assessment Method 2: Projects** Students' projects (in all courses in which projects are assigned) will be graded via two-part rubric that considers both design and technical expertise. We will track the grades for the technical expertise portion.

   1. **Success Criteria 1: Project Grades** An average grade of 80 or above on the technical expertise portion of a project will be considered a success.

C. **Assessment Method 3: Performance on External Coding Apps** Students will be encouraged to participate in "coding challenge" websites such as CodeWars or LeetCode.

   1. **Success Criteria 1: Tracking Progress** We will track student levels of engagement and accomplishment on the above platforms. A successful student will show continuous progress at solving problems. At least %80 of students must succeed to satisfy this success criteria.

II. **Goal 2: Design Expertise** Students will develop the ability to make the design decisions. When given a (possibly vague) idea for a project, students will be able to write a general design strategy for how to best implement the project idea. They will also be able to criticize, compare, and optimize different design strategies. They will recognize the difference between design and technical expertise.

A. **Assessment Method 1: Projects** Students' projects (in all courses in which projects are assigned) will be graded via two-part rubric that considers both design and technical expertise. We will track the grades for the design portion.

   1. **Success Criteria 1: Project Grades** An average grade of 80 or above on the design portion of a project will be considered a success.

III. **Goal 3: Theoretical Knowledge** Students will develop the mathematical and algorithmic theory necessary for an undergraduate level of understanding of computer science.

A. **Assessment Method 1: Grades in Discrete Mathematics, Algorithms, and Linear Algebra.**

   1. **Success Criteria 1** An average grade of below B will be considered a failure. An average grade of above 80 is a positive sign, but not necessarily an indication of success.

A. **Assessment Method 2 Qualitative Professorial Observation** Professors of upper-level CS courses will be on the lookout for indications of theoretical knowledge both in class discussions and in quality of turned in work.

1. **Success Criteria 1:** This success criteria is difficult to state quantitatively. By necessity it will be more of an "I know it when I see it" measure.

IV. **Goal 4: Student Buy-In** This includes campus-wide student enthusiasm/engagement as well as the academic resiliency of students who major in computer science.

A. **Assessment Method 1: CS131 Engagement** Since our intro course, CS131, is the gateway into both the major and interdisciplinary projects with other majors, CS131 will be the measuring stick for campus-wide computer science enthusiasm and engagement.

1. **Success Criteria 1: CS131 Enrollment** We will track enrollment numbers for CS131. Given the universally acknowledged importance of computer science, if our student learning outcomes are good then our major will have a good reputation and people will sign up for the intro course. Fewer than 15 students per fall semester will be cause for concern.
2. **Success Criteria 2: Major Declaration Rate** We want to see CS131 students becoming CS majors/minors at as high a rate as possible. We will track both CS132 enrollment (the course after CS131) and the rate at which CS131 students become CS majors/minors. Fewer than eight sophomore majors will be a bad sign. Fewer than eight CS132 students will be a bad sign. We will determine success criteria for minor numbers at a later date.

A. **Assessment Method 2: Major/Minor Retention** We will track the rate of graduating CS majors/minors.

1. **Success Criteria 1: Major/Minor head count** Fewer than 5 graduating majors will be a bad sign. The more we have, the better. Numbers for minors will be decided later.

V. **Goal 5: Professional Success** We will ask our majors to self-report what jobs offers they get and what graduate programs they are admitted to.

A. **Assessment Method 1: Proportion of Graduating Majors Continuing in Computer Science** We will track the number of graduating majors who receive (and accept) offers for CS jobs and graduate programs.

1. **Success Criteria 1:** All our majors who want to move on to pursue Computer Science should be able to find an opportunity to do so. If a non-trivial proportion of the ones who want to continue are unable to, that is a bad sign.

A. **Assessment Method 2: Internships** Summer internships are a valuable way for students to gain experience both in professional work environments and in job hunting. They also give students a chance to put what they've been learning into action and perhaps discover some of their own blind spots. We will track the rates at which our students get both internal and external interships.

    1. **Success Criteria 1:** We expect every CS major to have gotten some outside work experience by the time the graduate Wells. If more than a very few outliers are not successfully competing for such positions, that is a bad sign.