Computer Science Assessment Plan

0.1 Computer Science Mission Statement

Our mission is to produce students with both theoretical and technical expertise in the field of computer science. In every class our students take, they will learn how to formulate a goal, come up with a design plan to implement the goal, and then execute the plan successfully. These fulfill the Essential Skills 1, 2, and 4 of Wells College's student learning goals. Our students will also gain an appreciation for the abstract mathematical beauty of Computer Science. This reflects the Content Goals 1 and 2 of Wells College's student learning goals.

0.2 Computer Science Learning Goals

In this section we identify and define our learning goals. Note that for the purposes of this assessment plan, we find do not find the distinction between program goals and learning goals a useful one. Instead, we prefer the following structure:

1. LEARNING GOALS: What we want the students to do and/or become
2. ASSESSMENT METHOD: How we measure whether the students are fulfilling a particular learning goal.
3. SUCCESS CRITERIA: What the data collected by a particular assessment method has to be in order to indicate student success for a particular learning goal

The colleges learning goals will be used to inform our plans learning goals, since our plan does not include program 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. This learning goal corresponds to Essential Skill 1 of the college's learning goals.

Theoretical Knowledge

This learning goal refers to the mathematical theory behind Computer Science. Understanding why recursion works is an instance of theoretical knowledge. This learning goal corresponds to Content Goal 1 of the college's learning goals.

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 a technological platform. This learning goal corresponds to Essential Skills 1 and 4 of the college's learning goals.

Capacity for Self-Driven Learning

In recognition of the fact that the field of computer science is both always changing, this learning goal refers to the ability of students both to independently gain mastery of the skills they learn in the classroom and to seek out and learn beyond the scope of their class assignments.

0.3 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. **Goal 2: Design Expertise** Students will develop the ability to make 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.

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

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

1. **Assessment Method 1: Grades in the Algorithms class.**
    a. **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.
2. **Assessment Method 2 Effective Use of Math Pre-Reqs** Certain of our CS courses make heavy use of prerequisite mathematics courses. We qualitatively will assess the extent to which students can successfully leverage their mathematical knowledge in these courses.
    a. **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.

E. **Goal 4: Capacity for Self Driven Learning** Our students will have the ability to independently both learn new knowledge/skills and improve/strengthen the knowledge/skills they learn in class.

1. **Assessment Method 1: Performance on External Coding Apps** Students will be encouraged to participate in "coding challenge" websites such as CodeWars or LeetCode.
    a. **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.

1. **Assessment Method 2: Creative Solutions to Assignments or Projects**
    a. **Success Criteria 1: Tracking Progress** We will look for techniques or ideas in students' work that were not taught in class. If these are absent or rare it will indicate failure