



**2015-2016 CSCI EETF Assessment Year End Report, June, 2016**

| Program Name(s)       | EETF Faculty Rep | Department Chair |
|-----------------------|------------------|------------------|
| M.S. Computer Science | Matt Johnson     | Matt Johnson     |

[NOTE: Items A, B, C, and D are identical to your Page 2 on your Annual Report for CAPR. Please simply cut and paste from there. Item E is unique to the CSCI EETF.]

**A. Program Student Learning Outcomes**

Students graduating with an M.S. in Computer Science from CSU East Bay will be able to:

1. apply advanced computer science theory to computational problems
2. demonstrate advanced understanding of the mechanisms, components and architecture of current computing systems
3. apply emerging technologies and advanced algorithmic design
4. critique, plan and produce complex software applications
5. research and analyze current computer science literature

**B. Program Student Learning Outcome(s) Assessed**

Following our assessment plan, we are closing the loop on PLO #3 this year. The department does collect assessment data for all targeted courses each year, however, so as to track trajectories for scores on all PLOs.

Post-assessment quizzes were administered for three courses (addressing PLO #3):

- CS 6260 (Computational Complexity – Core Requirement) Developing PLO 1 and PLO 3
- CS 6560 (Operating Systems Design – Core requirement) Developing PLO 2 and PLO 3
- CS 6901 (Graduate Capstone) Mastering PLO 1, PLO 2, and PLO 3

**C. Summary of Assessment Process**

The Department created SLOs and PLOs for the Master in Computer Science program in the academic year 2012-2013. The Department made the decision to use Blackboard as a means to provide students with an assessment exam that addresses the SLOs of each course. The SLOs for each course have been mapped to the program PLOs and the ILOs of the university. The

assessment exams were developed for the required courses in the program, as well as a representative set of elective courses. The assessment instruments were then made available to the department faculty via a BlackBoard repository. Instructors teaching courses which were to be assessed in a given year deployed the tests and reported the results back to the Graduate Coordinator.

The Department has been using this assessment mechanism for three years now and can evaluate its advantages and disadvantages. Unfortunately, evaluating the results of the assessment exams as they stand is challenging, as each assessment contains questions addressing multiple PLOs. Due to a BlackBoard limitation, the results for individual PLOs cannot be automatically aggregated and compared across multiple courses, and instead must be tabulated by hand. To solve this problem, for the semester-based program, the Department agreed to develop assessment instruments that address only one PLO at a time. This will allow assessment to be automated, providing the opportunity to assess more courses, and assess those courses more frequently. The Department will be developing the new assessment instruments for the semester-based program beginning next year.

An additional challenge in the current system is assessing PLOs for both the Master's in Computer Science program and the Master's in Computer Network program. Since the programs share the great majority of the courses, but have different PLOs, it has been necessary to provide separate mappings of course SLOs to the PLOs of the two different programs, or to include additional questions on the assessment instruments to address the different PLOs. Fortunately, this difficulty will be eliminated under the semester-based program as the Master's in Computer Science and the Master's in Computer Networks have been combined into a single program with common PLOs.

In evaluating the assessment scores for PLO #3, we find very positive results in three of the four courses assessed. CS 6260, CS 6560, and CS 6901 are required courses for all students, except a small number of students who elect to complete a thesis rather than taking the comprehensive exams (CS 6901.) As a result, these assessment scores reflect the performance of the program as a whole in achieving PLO #3.

In regards to closing the loop and using the results of the assessment process to improve student learning for PLO #3, it would appear that the CS 6260 and CS 6560 need only fine tuning, while CS 6901 instructors could attempt to address consistency. In Fall 2015, CS 6901 students successfully demonstrated their proficiency regarding PLO #3, while Spring 2016 students were much less successful.

**CS 6260:**

This required course in Computational Complexity addresses material which is well-defined and theoretical in nature. It requires advanced understanding of algorithmic design, which is why it particularly addressed PLO #3. There is little room for addressing different or additional material, so fine-tuning would be aimed towards providing additional learning opportunities for students who were unsuccessfully served by the current class format. These opportunities might include high impact educational practices such as collaborative projects or swapped classrooms.

**CS 6560:**

This required course in Operating Systems Design addresses material which has a well-defined core of material but is constantly being enhanced by new research and advances in the industry. It clearly maps to both parts of PLO #3, in applying emerging technologies, and requiring advanced knowledge of algorithmic design. While perhaps two thirds of the course material is necessary for all students in the program, the remaining concepts may be presented through a number of avenues. Study of existing commercial operating systems, review of seminal research papers, and creation of software artifacts may all provide opportunities for learning the desired concepts. All of these avenues are included in the current incarnation of the course, with some students finding one or another more useful in providing a grasp of the necessary ideas. The instructor regularly selects or removes research papers for review as they prove more or less successful in delivering the needed material. Programming assignments are also modified or enhanced as required. This sort of fine tuning may provide for even better student success.

**CS 6901:**

This course comprises the capstone examinations, in which students take tests in 6 different areas of Computer Science. The material is well-defined and requires mastery of both theoretical and applied concepts. All material has been developed in earlier courses in the program, but here must be mastered outside of the structure of the earlier courses and shown to be current and comprehensive. To help prepare students for this evaluation, instructors typically run study groups throughout the quarter in which students solve problems from older exams, writing proofs, and designing software solutions to problems. As can be seen from the Fall 2015 results, students can successfully achieve the PLO goal via this mechanism. The poorer performance in Spring 2016 may be due to the composition of the study groups, the student need for different learning avenues, or another unknown factor. Since study groups are usually formed by the students themselves, it may be that less proficient students joined the same groups, and then did not benefit from the more experience of the more proficient students. This could be addressed by the instructor by assigning the students to groups, or by using BlackBoard functionality to allow groups to easily share their work. Wikis or blogs could also help support students who are less comfortable in the study group environment, and could learn more effectively using online tools. The Department will suggest these modifications to instructors in the coming year to enhance the learning opportunities for students, and attempt to provide consistently excellent

student performance on PLO #3.

#### **D. Summary of Assessment Results**

| <b>2015-2016</b> | <b>Assessment Results</b>               | <b>1</b> | <b>2</b> | <b>3</b> | <b>4</b> | <b>5</b> |
|------------------|---|----------|----------|----------|----------|----------|
|                  | CS 6260 Computational Complexity        | 7.8      |          | 8.7      |          |          |
|                  | CS 6560 Operating Systems Design        |          | 8.9      | 8.9      |          |          |
|                  | CS 6901 Graduate Capstone (Fall 2015)   | 6.7      | 6.8      | 8.2      |          |          |
|                  | CS 6901 Graduate Capstone (Spring 2016) | 4.6      | 8.1      | 6.4      |          |          |

#### **E. Suggestions and Recommendations for the CSCI EETF in the Future**

NONE