UNIVERSITY OF SAN FRANCISCO | College of Arts and Sciences

<Bachelor of Science/Computer Science/Major>

ASSESSMENT REPORT
ACADEMIC YEAR 2017 – 2018

I. LOGISTICS & PROGRAM LEARNING OUTCOMES

1. **Please indicate the name and email of the program contact person to whom feedback should be sent (usually Chair, Program Director, or Faculty Assessment Coordinator).**

   EJ Jung, ejung2@usfca.edu, Faculty Assessment Coordinator of CS dept.
   Dave Wolber, wolberd@usfca.edu, Chairperson of CS dept.

2. **Were any changes made to the program mission statement since the last assessment cycle in October 2017? Kindly state "Yes" or "No." Please provide the current mission statement below. If you are submitting an aggregate report, please provide the current mission statements of both the major and the minor program.**

   No changes were made.

   Students who graduate with a Bachelor of Science (B.S.) degree in Computer Science will be prepared for both graduate school and for software development careers. The curriculum provides a solid base in computer science fundamentals that includes software design and development, problem solving and debugging, theoretical and mathematical foundations, computer systems, and system software.

3. **Were any changes made to the program learning outcomes (PLOs) since the last assessment cycle in October 2017? Kindly state "Yes" or "No." Please provide the current PLOs below. If you are submitting an aggregate report, please provide the current PLOs for both the major and the minor programs.**

   No changes were made.

   - THEORY: Explain and analyze standard computer science algorithms and describe and analyze theoretical aspects of various programming languages.

- APPLICATION: Apply problem-solving skills to implement medium- and large- scale programs in a variety of programming languages.
- SYSTEMS: Describe the interactions between low-level hardware, operating systems, and applications.
- PROJECT: Demonstrate effective communication and organization as part of a team of software developers or researchers collaborating on a large computer program.

4. **Which particular Program Learning Outcome(s) did you assess for the academic year 2017-2018?**
THEORY: Explain and analyze standard computer science algorithms and describe and analyze theoretical aspects of various programming languages. In particular, we focused on the first part of the THEORY requirement, "explain and analyze standard computer science algorithms."

II. METHODOLOGY

5. **Describe the methodology that you used to assess the PLO(s).**

We used direct methods to assess this learning outcome by using the grades of CS 245 Data Structures and Algorithms whose learning outcome is to learn standard computer science algorithms and learn to analyze their time complexities. This is a required course for CS major and also a prerequisite for many upper-division courses. Thus mastery in the course material is highly important for students' timely graduation and also success in the upper-division courses. The final grade is calculated over written assignments (10%), programming projects (40%), and exams (50%) to demonstrate their mastery in various ways.

III. RESULTS & MAJOR FINDINGS

6. **What are the major takeaways from your assessment exercise?**
This section is for you to highlight the results of the exercise. Pertinent information here would include:
   a. **how well students mastered the outcome at the level they were intended to,**
   b. **any trends noticed over the past few assessment cycles, and**
   c. **the levels at which students mastered the outcome based on the rubric used.**

Level	**Percentage of Students**
Complete Mastery of the outcome (A)	18% (7/38)
Mastered the outcome in most parts (B, B+, A-)	45% (17/38)
Mastered some parts of the outcome (C, C+, B-)	32% (12/38)

Did not master the outcome at the level intended (C- and below, considered as fail in CS major)	5% (2/38)

95% of the students demonstrated mastery at least in some parts of the outcome, and 63% of the students did in most parts.

IV. CLOSING THE LOOP

7. Based on your results, what changes/modifications are you planning in order to achieve the desired level of mastery in the assessed learning outcome? This section could also address more long-term planning that your department/program is considering and does not require that any changes need to be implemented in the next academic year itself.

While most students mastered at least some parts of the outcome, two students who did not master the outcome at the intended level and failed the course. Both of them frequently missed the classes and/or showed little engagement during the class. One of them is retaking this course in Fall 2018. The other student is currently not taking any CS courses. The student who is continuing in CS major had coaching sessions with the CASA academic coach on the time management and seemed to have benefited from them. We intend to work more closely with the CASA academic coaches for the struggling students in the future.

8. What were the most important suggestions/feedback from the FDCD on your last assessment report (for academic year 2016-2017, submitted in October 2017)? How did you incorporate or address the suggestion(s) in this report?

The FDCD noticed some of our courses did not map to any PLOs in our curricular map. In fact they did, but we only showed how a category of courses map into a PLO instead of individual courses. In this submission, we revised the map to show how individual courses map to PLOs.

ADDITIONAL MATERIALS

(Any rubrics used for assessment, relevant tables, charts and figures should be included here)

Revised PLOxCurricular map is attached.

	PLO1	PLO2	PLO3	PLO4
Program Learning Outcomes X Courses	THEORY: Explain and analyze standard computer science algorithms and describe and analyze theoretical aspects of various programming languages.	APPLICATION: Apply problem-solving skills to implement medium- and large- scale programs in a variety of programming languages.	SYSTEMS: Describe the interactions between low-level hardware, operating systems, and applications.	PROJECT: Demonstrate effective communication and organization as part of a team of software developers or researchers collaborating on a large computer program.
Courses or Program Requirement				
110: Introduction to Computer Science I	I	I	I	I
112: Introduction to Computer Science I	I	D	I	I
212: Software Development	D	D		D
SYSTEMS:				
220: Introduction to Parallel Programming			D	
221: C and Systems Programming			D	
245: Data Structures and Algorithms	D	D		D
315: Computer Architecture			M	
326: Operating Systems			M	
THEORY:				
345: Programming Language Paradigms	M			
411: Automata Theory	M			
414: Compilers	M			
APPLICATIONS: Any 300 and 400 level course except for except 398 or 498. Examples below		M		
333: Introduction to Database Systems		M		
336: Computer Networks		M		
360: Data Visualization		M		
419: Computer Graphics		M		
420: Game Engineering		M		
451: Data Mining		M		
480: Computers and Society		M		
398/498: Directed Reading and Research		D		D
490: Senior Team Project		M		M
	Key:			
	I = Introductory			
	D = Developing			
	M = Mastery			