UNIVERSITY OF SAN FRANCISCO | College of Arts and Sciences

<MS in Computer Science Bridge Program>

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.**

This is the first year that we submit the assessment report on the MS in CS Bridge program. The mission of the MS in Computer Science Bridge program has an additional goal of preparing students from non-CS background ready for the MSCS program in addition to the mission of the MSCS program.

The mission of the MS in Computer Science Bridge program is:

To prepare students for Master's in Computer Science at USF who are changing fields from non-computer science backgrounds and to give students who do not have a computer science background enough knowledge to do basic software development.

To provide students a strong theoretical background in computer science and deep technical programming skills by focusing on one-on-one student interaction and fostering the unique capabilities of each student.

Our mission statement coincides with the university mission to give students the knowledge and skills needed to succeed as professionals, and we are sensitive to the needs of our extremely diverse student population.

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.**

This is the first year that we submit the assessment report on the MS in CS Bridge program. The program learning outcomes of the MS in CS Bridge program has additional PLOs for the first year before students start the MS program.

Students who pass the first year of the bridge program and proceed to the MS in Computer Science will be able to:

- Application: Apply problem-solving skills to implement medium- and large-scale programs in a variety of programming languages.
- Theory: Explain and analyze standard computer science algorithms
- Systems: Describe the interactions between low-level hardware, operating systems, and applications

Students who graduate with a MS in Computer Science will be able to:

- Demonstrate advanced knowledge in a breadth of topics in computer science, including theory, systems, and development.
- Demonstrate mastery in at least one area of specialization in computer science.
- Demonstrate ability to independently solve advanced problems in academia or industry.
- Demonstrate ability to learn, use, and adapt emerging developments in the state-of-the-art in computer science.

4. **Which particular Program Learning Outcome(s) did you assess for the academic year 2017-2018?**

Theory: 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 545 Data Structures and Algorithms where students learn standard computer science algorithms and learn to analyze their time complexities. This is a required course for the MS in CS Bridge program. Mastery in cs545 is defined by achieving the learning outcomes listed below. The students who successfully pass this class should be able to do the following:

- Analyze running times of algorithms, including analyzing recursive code using recurrence relations (assessed by homework assignments and exams)

- Write significantly sized Java programs with complex logic (assessed by projects)
- Have a deeper understanding of recursion, and be able to implement several standard tree and graph algorithms recursively (assessed by projects and exams)
- Implement interfaces in Java, including Iterator, Comparable, Comparator (assessed by projects)
- Understand the following data structures and be able to implement the corresponding algorithms: Stacks, Queues, and Lists; Binary Search Trees; Skip Lists; General Trees; Tries for String Matching; Hash Tables; Heaps, Binomial Heaps; Sorting algorithms; Disjoint Sets; Graphs; B-Trees; Dynamic programming; NP-completeness. (assessed by homework assignments, exams and projects)

The final grade for this course is computed based on the following breakdown:

Assignment Group	Weight
Homework	10%
Projects	40%
Midterm 1	15%
Midterm 2	15%
Final Exam	20%

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.

To assess mastery, we split the students into four groups:

1. "complete mastery of the outcome"
2. "mastered the outcome in most parts",
3. "mastered some parts of the outcome"
4. "did not master the outcome at the level intended."

In the table below, we listed how many students fell into each category for each assignment group in CS545:

Description	Type	Complete Mastery	Mastery in Most parts	Mastery in Some parts	Did not master at the intended level
Homeworks	Direct	8	8	3	0
Project 1	Direct	19	0	0	0
Project 2	Direct	18	0	1	0
Project 3	Direct	18	0	0	1
Project 4	Direct	15	3	1	0
Project 5	Direct	16	2	1	0
Midterm 1	Direct	10	3	5	1
Midterm 2	Direct	6	8	4	1
Final Exam	Direct	10	6	3	0
Final Grade	Direct	12	7	0	0

Two students had some attendance issues that led to less than complete mastery of the discussion work. Research paper reviews trended upward over the course of the semester: early on, students tended to only summarize the work rather than critiquing it. Towards the end of the semester, they were better at identifying weaknesses in the work and suggesting alternative approaches. For projects, students were allowed to correct their work and resubmit for half credit back. This took some extra time but improved the number of students reaching complete mastery.

This is the first year that we evaluated this particular learning outcome, so we have not found any trends yet.

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.**

No students in the class fell in the "Mastery in Some parts" or "Below the intended level" categories at the end of the semester (based on the overall score computed according to the grade breakdown listed above). Students did very well in the final exam, with only three students getting less than a "B". Overall, students did better on the final exam compared to the midterm exams.

The instructor conducted mini coding "boot camps" to prepare students for the exams, and this approach seemed to have helped.

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?**

N/A (This is the first year that we submit the assessment report on the MS in CS Bridge program.)

ADDITIONAL MATERIALS

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

As this is the first assessment report, the PLOxILO map and the PLOxCurricular map are attached. Below we provide grading rubrics for two assignments in cs545: homework 1 and project 4:

CS545 Homework 1, Algorithm Analysis

This homework was on Algorithm Analysis, and was used to partially assess the first learning outcome listed in section 1.

Grading rubric, Total points: 15 pts

Problem 1 (3 pts): Ranking of functions based on the growth rate and explanation.

Problem 2 (3 pts) Proving big O and big Omega bounds.

Problem 3 (9 pts): Analyzing running time of non-recursive piece of code

(3a) (1.5 pts) Analyzing running time of code that contains a simple loop.

(3b) (1.5 pts) Proving the Theta bound.

(3c) (2pts) Analyzing running time of code that contains nested loops, and the index of the inner loop depends on the index of the outer loop.

(3d) (2 pts) Analyzing running time of code that contains nested loops, where the index of the nested loop grows exponentially.

(3e) (2 pt) Analyzing running time of the code that includes loops and function calls within a loop.

CS545 Project 4, Sorting

In this project, students were asked to implement a variety of sorting algorithms. This project was used to partially assess the second learning outcome and partially the fifth learning outcome listed in section 1.

Project 4 Grading Rubric

Total 90 points

1. Passing JUnit Tests: 50 pts

testInsertionSort 5 pts

testIterativeMergeSort 6pt

testExternalSort 13 pts

testHeapSort 9pts

testBucketSort 9pts

testRandomizedQuicksort 4 pts

testHybrid 4 pts

2. Code: 40 pts

Insertion sort 4 pts

Iterative merge sort 5 pts

In-place Heap Sort 8 pts

Building the heap (from the bottom up) 2 pt

Calling removeMax repeatedly and swapping with correct elem 2 pt

Fixing the heap after removing max: 2 pt

Dealing with low, high correctly: 2pt

Randomized Quick Sort 2 pts

Hybrid sort (that uses both quicksort and insertion sort) 1.5 pts

Test file + Readme for the Hybrid sort (that compare it with randomized quick sort) 1.5 pt

Bucket sort 7 pts

Iterating over elements, computing the index of the bucket for each element 1.5 pt

Inserting element into the correct place in the LinkedList 3 pt (inserting 1pt + in sorted order 2pt)

Iterating over the bucket array and writing elements back to the original array 1.5pt

Handling low and high correctly 1pt

External sort 11 pts

Reading the large file chunk by chunk (each chunk has no more than k integers) 1.5pt

Saving each chunk into an array, sorting it and writing to a temp file 3.5 pt

Reading all temp files and merging them into a large output file 6 pt

	PLO1	PLO2	PLO3
Institutional Learning Outcomes X Program Learning Outcomes	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.
Institutional Learning Outcomes			
1. Students reflect on and analyze their attitudes, beliefs, values, and assumptions about diverse communities and cultures and contribute to the common good.			
2. Students explain and apply disciplinary concepts, practices, and ethics of their chosen academic discipline in diverse communities.		X	
3. Students construct, interpret, analyze, and evaluate information and ideas derived from a multitude of sources.	X	X	X
4. Students communicate effectively in written and oral forms to interact within their personal and professional communities.			
5. Students use technology to access and communicate information in their personal and professional lives.	X	X	X
6. Students use multiple methods of inquiry and research processes to answer questions and solve problems.	X	X	X
7. Students describe, analyze, and evaluate global interconnectedness in social, economic, environmental and political systems that shape diverse groups within the San Francisco Bay Area and the world.			

	PLO1	PLO2	PLO3
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.
Courses or Program Requirement			
CS 514 Accelerated Object Oriented Programming	I/D	D	
SYSTEMS:			
CS 520 Introduction to Parallel Computing		D	D
THEORY:			
Math 501 Discrete Mathematics	D		
CS 545 Data Structures and Algorithms	M	D	
APPLICATIONS:			
CS Elective		M	
CS Practicum: Practical Industry or Research Experience		M	
	Key:		
	I = Introductory		
	D = Developing		
	M = Mastery		