William Paterson University
College of Science and Health - Department of Computer Science

Fall 2015 – Spring 2017 Assessment Cycle
Analysis of the Program's Student Outcome Assessment Data

Program's Student Outcome:

S2: Demonstrate abilities to apply knowledge of mathematics to the discipline of computer science.

ABET's Related Student Outcomes: (a), (i).

Assessment Committee Members: Gilbert Ndjatou (Chair), Bogong Su, Erh-Wen Hu

Date: November 21, 2017

Updated On: January 30, 2018

A. Analysis of the Assessment Data

For the assessment period Fall 2015 to Spring 2017, this student outcome was assessed in the following four courses: CS2600, CS2800, and CS3420. The data of CS2600 were collected in Fall 2016 and Spring 2017. The data of CS2800 were collected in Fall 2015, Spring 2016, Fall 2016 and Spring 2017. The data of CS 3420 were collected in Spring 2016, Fall 2016 and Spring 2017.

In CS 2600, 24 students took in Fall 2016 and all of them achieved adequate ability. In Spring 2017, 8 students out of 24 (33%) had less than adequate ability. Two of the students who failed this course had more than 10 absences. One student who failed the course only handed in one homework out of four. It is noticed that the quizzes really kept their knowledge up-to-date instead of waiting until exam times in both semesters. We therefore suggest that this practice should be continued in the future.
In CS2800, 16, 24, 23, and 24 students respectively took the course and 11 students out of them (12.6%) had less than adequate ability. It is observed that few students have weak mathematic background.
In CS3420, 19, 19, and 15 students respectively took the course and 4 students out of them (7.5%) had less than adequate ability. It is noticed that CS 2600 and CS 3420 are closely related.

B. Suggestions for Improvement

- In CS2600, emphasizing more on the application of discrete mathematics and giving more quizzes or class exercises to let students practice.
- In CS 3420, the instructors of CS2600 and CS3420 must have close and frequent interactions. Animations can be very helpful in helping students to learn certain algorithms.

C. Improvement Implemented

Spring 2018

D. List all the "performance level/frequency/percentage" tables and their sources.

a. Faculty Course Assessment Report: CS 2600, Fall 2016

Data Collected: Each student's level of performance on quizzes, the midterm exam the final exam, and homework assignments.
Method of Collection: Nine (9) close-book quizzes, a midterm exam, a final exam and 5 homework assignments were given. Their relative weights were 80% exam-work and 20% homework assignment. The exams include the topics in 1) logics 2) mathematical proofs 3) set theory and functions 4) relations 6) counting and probability

Performance Levels	Frequency	Percentage
No Ability	0	0%
Some Ability	0	0%
Adequate Ability	10	42%
More than Adequate Ability	7	29%
High Ability	7	29 %

Observations: It is delighted that no one was left behind in CS2600 this semester. I think it was the weekly quizzes that made this happen. All the quizzes were designed as a review of the lectures of the previous week. Selected questions were not too hard but would cover every key point in the previous week. It showed that students' grades in exams are highly related to their performances in quizzes. The weekly quizzes enhanced students' memory results by forcing them to do in-time reviews right after the lectures. The feedbacks from students also indicate that the quizzes helped them a lot when they were preparing for the exams.
The mathematical proofs were the most challenging topics for students. I spent one or two weeks more than I expected on those topics. The good thing was that most of the students could handle the basic steps of proofs at last. If I could teach those topics again, I would only focus on one kind of problems in number theory to show them how to give a valid and concise proof.

In summary, it has been a successful semester for me. It is glad to see that students understand the basic mathematical concepts and can apply those concepts to different problems.

---

b. Faculty Course Assessment Report: CS 2600, Spring 2017

Data Collected: Each student's level of performance on the discrete mathematical concepts and skills based on homework, quizzes, and exam scores.
Method of Collection: The assessment of the performance levels is based on the scores of 4 homework sets, 6 quizzes (only the highest 4 scores counted), 2 exams, and 1 final cumulative exam. The final exam was comprehensive (i.e., it covered all the materials taught in the entire semester, with more questions coming from the materials after the second exam).

Performance Levels	Frequency	Percentage
No Ability (Level of performance of F)	4	17%
Some Ability (Level of performance of D)	4	17%
Adequate Ability (Level of performance of C)	5	21%
More than Adequate Ability (Level of performance of B)	3	12%
High Ability (Level of performance of A)	8	33%

Observations: The overall performance of this class was pretty good. One third of the students were at the A level of performance and 83% of the students passed this class. In the past, I gave many homework (for example, in the spring 2009 semester, I gave 12 homework sets) but this semester, I only gave them 4 homework sets. Students have the tendency to copy each other's homework. Instead of homework, I decided to give them 6 quizzes (the lowest score of 2 quizzes were dropped) almost every other week. The quizzes really kept their knowledge up-to-date instead of waiting until exam times. Two of the students who failed this course had more than 10 absences. One student who failed the course only handed in one homework out of four.
As usual, I emphasized on the application of discrete mathematics as it relates to data structures, digital logics, databases, programming, and artificial intelligence. I used many examples to illustrate the concepts even though they might not know anything about digital logics, databases, or artificial intelligence. Students seemed to be interested in the applications of discrete mathematics.

---

c. Faculty Course Assessment Report: CS 2800, Fall 2015

Data Collected: Each student's level of homework and test.
Method of Collection: Each student was given an exam on the purely numerical aspects of binary numbers, their representation, and binary computation.

Performance Levels	Frequency	Percentage
Some Ability	2	13 %
Adequate Ability	3	19 %
More than Adequate Ability	3	19%
High Ability	8	50 %

Observations: Most students have adequate ability, however few students have weak mathematic background this semester.

---

d. Faculty Course Assessment Report: CS 2800, Spring 2016

Data Collected: Each student's level of homework and test.
Method of Collection: There were three homework (30%) and one test (70%)

Performance Levels	Frequency	Percentage
No Ability	1	4%
Some Ability	1	4 %
Adequate Ability	8	33 %
More than Adequate Ability	5	21%
High Ability	9	38 %

Observations: Most students have adequate ability, however few students have weak mathematic background this semester.

---

e. Faculty Course Assessment Report: CS 2800, Fall 2016

Data Collected: Each student's level of homework and test.
Method of Collection: There were three homework (30%) and one test (70%)

Performance Levels	Frequency	Percentage
No Ability	2	9 %
Some Ability	0	0 %
Adequate Ability	8	35 %
More than Adequate Ability	5	22%
High Ability	8	35 %

Observations: Most students have adequate ability, however few students have weak mathematic background this semester. The statistical data are a little bit worse than S16 semester.

---

f. Faculty Course Assessment Report: CS 2800, Spring 2017

Data Collected: Each student's level of homework and test.
Method of Collection: There were three homework (30%) and one test (70%)

Performance Levels	Frequency	Percentage
No Ability	1	4%
Some Ability	4	17%
Adequate Ability	3	13%
More than Adequate Ability	7	29%
High Ability	9	38%

Observations: Most students have adequate ability, the percentage of "High Ability" and "More than Adequate Ability" are more than previous semesters; however, few students have weak mathematic background.

---

g. Faculty Course Assessment Report: CS 3420, Spring 2016

Data collected: Each student is given a score on the three tests, final exam, and all programming projects.

Method of collection programming assignments, quizzes, and the final exam targeting the assessment of students' ability to apply mathematics to the discipline of computer science. Math-related topics include concept of abstraction (e.g., ADT), number theory (e.g., modulo arithmetic), functions (e.g., hash functions), combinatorics (e.g., queuing), algebra (e.g., matrices), sets, maps, recursion (e.g., used various algorithms), and Big-O analysis of algorithms.

Performance Levels	**Frequency**	**Percentage**
No Ability (level of performance of F)	1	5.2
Some Ability (level of performance of D)	0	0
Adequate Ability (level of performance C)	11	57.9
More than Adequate Ability (level of performance of B)	4	21.1
High Ability level of performance of A)	3	15.8

Observation:

Make it a recurring theme throughout the course that there is close relationship between mathematics as there were significant number of students who failed to connect the topics such as analysis of algorithms covered in this class to what then have learned about Big-O notation in the prerequisite CS2600 Discrete Structures class.

It is recommended that the characteristics of algebraic functions and their operations – particularly the logarithmic functions – be thoroughly reviewed. Logarithmic function is important not only in analyzing algorithm; it is also important in understanding a why a balanced binary search tree (BST) is an efficient storage structure in searching.

To improve the learning outcomes for both CS2600 and CS3420, the instructors of both classes must have close and frequent interactions. Interestingly, there has been quite some discussion on whether discrete math should be a prerequisite for data structures or the other way around. As it turns out, it could go either way. For example, here at William Paterson, we teach discrete structures first while in other schools such as Duke University, CS majors take data structures ahead of discrete math.

h. Faculty Course Assessment Report: CS 3420, Fall 2016

Data collected: Each student is given a score on the three tests, final exam, and all programming projects.

Method of collection programming assignments, quizzes, and the final exam targeting the assessment of students' ability to apply mathematics to the discipline of computer science. Math-related topics include concept of abstraction (e.g., ADT), number theory (e.g., modulo arithmetic), functions (e.g., hash functions), combinatorics (e.g., queuing), algebra (e.g., matrices), sets, maps, recursion (e.g., used various algorithms), and Big-O analysis of algorithms.

Performance Levels	**Frequency**	**Percentage**
No Ability (level of performance of F)	1	12.5
Some Ability (level of performance of D)	0	0
Adequate Ability (level of performance C)	11	33.3
More than Adequate Ability (level of performance of B)	4	37.5
High Ability level of performance of A)	3	16.7

Observation:

1. Big-O analysis of algorithms always presents a challenge to most students. Therefore, the instructor introduced the topic of Big-O early in the course and used it throughout the semester to assess the efficiency of all algorithms. In the beginning, many students struggled with it but eventually, most students were able to use it to analyze and compare the efficiency of algorithms used in most data structures.

2. Most students find the programming assignment on measuring and comparing the execution time as a function of n (data size or instruction count) for certain algorithms such as various sorting a list of objects. Before this assignment, students care only the correctness of the their program. After this assignment, they realize that there is a world of difference between bubble-sort and quicksort or other fast sort algorithms.

3. As the quantitative stats in the above table indicates, only one student fail the class and more than half of the class achieved "More than Adequate Ability" and "High Ability".

i. Faculty Course Assessment Report: CS 3420, Spring 2017

Data collected: Each student is given a score on the three tests, final exam, and all programming projects.

Method of collection programming assignments, quizzes, and the final exam targeting the assessment of students' ability to apply mathematics to the discipline of computer science. Math-related topics include concept of abstraction (e.g., ADT), number theory (e.g., modulo arithmetic), functions (e.g., hash functions), combinatorics (e.g., queuing), algebra (e.g., matrices), sets, maps, recursion (e.g., used various algorithms), and Big-O analysis of algorithms.

Performance Levels	Frequency	Percentage
No Ability	0	0.0%
Some Ability	2	33.3%
Adequate Ability	1	20.0%
More than Adequate Ability	8	40.0%
High Ability	4	6.7%

Observations:

Animations can be very helpful in helping students to learn certain algorithms such as sorting (e.g., quicksort) a list of object, balancing a binary search tree, or heaifying a heap. They should be used as much as possible.