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

Fall 2018 – Spring 2019 Assessment Cycle
Analysis of the Results of the Evaluations of the Assessment Data
of
the Program Student Outcome

Program Student Outcome:

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

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

Date: May 28, 2019

Updated On: _____

A. Analysis of the Results of the Evaluations of the Assessment Data

For the assessment period Fall 2018 to Spring 2019, this student outcome was assessed in the following four courses: CS2600, CS2800, CS 3410, and CS 3420.

In CS 2600, the small (<10%) number of F's due to inattention, absence is more an issue of individual psychology/motivation, attitude, and social factors, neither pedagogical, nor curricular changes. Teaching more or doing more exercises, even changing relevance or application-promotion will not help if a student is absent or does not complete the work. On possible solution is to require pre-calculus (Math 1160) be completed before CS 2600 (prerequisite) or concurrently (co-requisite). In that manner, students will have a stronger mathematical background before CS 2600 Discrete Structures. The foundational knowledge and problem-solving skills will make the transition into Discrete Structures more uniform and less a learning curve. This is especially true in topics like functions and relations where requisite skill sets occur.

However, further analysis of CS 2600 data (and collecting more data) is required to determine if such an augmentation in prerequisite might lead to delaying graduation. Such factors are not inferable from the above statistics. With transfer students, the delay to graduation can be more pronounced (i.e. students with CS1 without Math 1150 College Algebra). Perhaps increasing credits to 4 credits (entailing greater background review) can rectify background differential in these multi-modal circumstances. An alternate solution is more early review tests and dropping students who would otherwise drift (have absences due to background deficiencies). To some degree, these solutions would benefit the "Some Ability" and to a lesser extent "Adequate Ability". Before making such changes, more data on CS 2600 is needed in this cycle, to determine the scope and magnitude of benefit.

CS 2800 is doing fine in terms of S2. No change is needed; a statistic of 0 at the lower tail indicates a very positive situation.

CS 3410 (Digital Logic) is a variable situation in that the problem of Fall 2018 has disappeared in Spring 2019. Whether the problem of Fall 2018 was just a transient pathological situation (induced by statistical variance) OR the problem was addressed by appropriate counter-measures, the problem is gone. Now, we need to collect more semesters of data to confirm the problem is gone permanently.

CS 3420 has had 0 in the "No Ability" category for both semesters considered, which means we have resolved the drifting student problem in prior courses; this is a positive indicator. However, a fluctuation in 30% at the "Some Ability" level suggests a high variance. We will need to collect more statistics and in particular, determine causes. We cannot increase credit (for more background) as we already reached 4 credits (there are no 5 credit courses). An alternative is to require Math 1160 Pre-Calculus prior to CS 3240 but this may delay graduation (or at least reduce timeliness thereof).
Part of the solution is to be less formal in the analysis of algorithms (ie. determine complexity class in a more informal manner than by inequality proof/manipulation in every case for Big-Oh analysis). Yes, we can apply L'Hospital's Rule or do extensive asymptotic analyses (it's fun as Polya would have it) but shouldn't that require at least Calculus 1 to apply properly; furthermore, would this really be beneficial to Computer Science students in the long term? We need to determine the level of rigor in the exact proof methodology. Analysis can be done more informally, using pre-existing theorems rather than bare inequality algebra (like the situation with limits, whether to go down to epsilon-delta level every time). More time and effort should be focused on algorithmics of the data structures (determining optimal solutions). This is especially true since we removed the CS 3720 Algorithms as a requirement (in response to ACM modifications in topical coverage/stresses).

B. Suggestions for Improvement

In each course, potential improvements are suggested above (in section A) in their respective contexts/courses.

C. Improvement Implemented

At this present stage of assessment, it is too early in the cycle to implement solutions, nay, even to suggest them; it data is inconclusive. Therefore, it would be even more pre-mature to assess and draw conclusions. Especially for CS 3420, we need to monitor the nature of deficiencies more exactly to determine demographics causalities and formulate solutions based upon the constituent student multi-modal populations.

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

a. **Faculty Course Assessment Report : CS2600, Fall 2018**

Data Collected: Each student's level of performance on the discrete mathematics concepts and skills are based on take-home assignments, quizzes, classroom performance, and test scores.
Method of Collection: 14 hands – on worksheets, 7 quizzes, 3 tests, a comprehensive final exam.

The class consisted of 20 students.

Performance Levels	Frequency	Percentage
No Ability (level of performance of F)	2	10%
Some Ability (level of performance of D)	1	5%
Adequate Ability (level of performance C)	10	50%
More than Adequate Ability (level of performance of B)	6	30%
High Ability level of performance of A)	1	5%

Observations: The spring semester is sometimes a difficult semester for all concerned. The frequency of storms and class cancellations made maintaining momentum in the class quite difficult. However, we persevered using take home quizzes instead of in class quizzes to make up class time. A review packet was also distributed at the end of the semester to help solidify the material covered in the course.

About half the students in the class performed at the level expected of college students. The other half skipped classes, failed to complete assignments, and did not do as well as could be expected on the final exam.

It is also true that students did not give up on the course. They were aware of the fact that if their final exam grade was better than their lowest test score, the final would replace that score, or a missing test score if they had one. This gave them incentive to keep working and some of them were able to recover.

There were some very good students in the class who did enjoy the material, particularly the material on graph theory. They were hoping that we would be able to include some enrichment material but the semester did not lend itself to this.

The textbook is a good one and readable for students. They also like the worksheet format where we solve typical problems in class before sending them home to do the rest. I do feel that we could have done more group work had the class not been canceled several times.

This class was held in a regular classroom as opposed to a computer lab. For this course, the difference was remarkable. I would not have wanted to teach this particular group of students in a room with available distractions.

NOTE: Please comment or provide some justifications for students having a performance level of F.
The two students whose grades were F grades had low attendance, missed assignments, tests and quizzes and failed to make them up. While the semester was somewhat fragmented by snowstorms and class cancellations, material was covered sometimes with take-home assignments and quizzes. The instructor was in touch electronically. These two students did not respond. One of them attempted to do some make-up the last two weeks of class and had no idea what was going on.

One of these students was going through a difficult time with her family and missed four weeks in a row. She then returned to class and after about a week, was missing for two weeks. She was unable to recover from these large gaps in the course material even though the topics are not always sequential. She might have benefitted from a withdrawal in the course, but did not want to do this.

The other of these students missed three quizzes and a test. He missed approximately one third of the class time during the semester. If there was a take home assignment he would miss class and show up at the instructor's office to turn in the assignment late, or not do the assignment at all. Repeated emails and suggestions to meet with tutors went unheeded. He attempted to make up all missed work at the end of the semester with a request for extra credit. It is clearly stated in the syllabus that this will not be allowed. Hopefully this experience has contributed to the maturation process for this individual.

b. Faculty Course Assessment Report : CS2600, Spring 2019

Data Collected: Each student's level of performance on quizzes, the midterm exam and the final exam.

Method of Collection: Six (6) quizzes, a midterm exam, a final exam were given and six homework assignments. Their relative weights were 20% quizzes, 30% midterm, 30% final and 20% homework. The exams include the topics in 1) logics 2) mathematical proofs 3) set theory and functions 4) elementary concepts of number theory 5) sequences, summation, and recursive definition and algorithms 6) relations 7) counting and probability 8) graph theory

Performance Levels	Frequency	Percentage
No Ability (Level of performance of F)	2	8.3%
Some Ability (Level of performance of D)	3	16.7%
Adequate Ability (Level of performance of C)	11	41.7%
More than Adequate Ability (Level of performance of B)	4	16.7%
High Ability (Level of performance of A)	4	16.7%

Observations:
One failed the course because of missing the final exam. Six (20.8%) students has less than adequate ability. Most of the CIT students' math background are weak. It took them a longer time to digest what learned in class. At the same time, the contents are numerous. I can only introduce the basic concepts and did not have enough time to give plenty of exercises. The mathematical proofs were the most challenging topics for students. I think discrete structure should have one more credit hour.

c. **Faculty Course Assessment Report : CS2800, Fall 2018**

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	0	0%
Some Ability	1	6%
Adequate Ability	7	39%
More than Adequate Ability	2	11%
High Ability	8	44%

Observations: Almost half of the students do not have adequate ability, the percentage of "High Ability" and "More than Adequate Ability" are much less than previous semesters; many students have weak mathematical background.

d. **Faculty Course Assessment Report : CS2800, Spring 2019**

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	0	0%
Some Ability	1	5%
Adequate Ability	7	35%
More than Adequate Ability	8	40%
High Ability	4	20%

Observations: Most students have adequate ability, showing their strong math background .

e. **Faculty Course Assessment Report : CS3410, Fall 2018**

Data Collected: Each student's level of homework and test.

Method of Collection: There were two homework and one test

Performance Levels	Frequency	Percentage
No Ability	4	19%
Some Ability	1	5%
Adequate Ability	5	24%
More than Adequate Ability	6	29%
High Ability	5	24%

Observations: Almost half students do not have adequate ability, the percentage of "High Ability" and "More than Adequate Ability" are much less than previous semesters; many students have weak mathematic background. Some students did not submit homework 2 and failed test 1. One question in final exam is the calculation of memory address bits from the given memory size, which is a simple logarithmic relationship; however more than 50 % students have wrong answers, even similar homework question was given and went over before exam.

f. **Faculty Course Assessment Report : CS3410, Spring 2019**

Data Collected: Each student's level of homework and test.

Method of Collection: There were two homework and one test

Performance Levels	Frequency	Percentage
No Ability	1	7%
Some Ability	0	0%
Adequate Ability	1	7%
More than Adequate Ability	5	36%
High Ability	7	50%

Observations: Most students have adequate mathematics ability, the percentage of "High Ability" and "More than Adequate Ability" are more than previous semesters; However one student, who failed last semester and retake this semester, still failed the first test.

g. **Faculty Course Assessment Report : CS3420, Fall 2018**

Data Collected: Performance based on students' ability to discrete mathematics to the discipline of computer science, particularly their ability to derive and interpret the Big-O expressions to estimate and compare the complexities of algorithms used in various data structures.

Method of Collection: Each student is given a score on each of the three tests (40%), the final exam, (25%), and 12 programming projects and homework assignments (35%).

Performance Levels	Frequency	Percentage
No Ability (F)	0	0%
Some Ability (D)	0	0%

Adequate Ability (C)	7	54%
More than Adequate Ability (B)	4	31%
High Ability (A)	2	15%

Observations: To learn Big-O analysis of algorithms proved times and again a challenge to many students. As the shown in the above table, less than half (46%) of the class achieved "More than adequate" or "High ability". A thorough review of algebra and arithmetic on topics including logarithms and arithmetic series proved to be necessary. Introduce the analysis of algorithm early in the course and apply it throughout the course also proved to be helpful.

h. **Faculty Course Assessment Report : CS3420, Spring 2019**

Data Collected: Students' ability to apply mathematics (mainly algebra) to the discipline of computer science, particularly their ability to use the needed math in assessing and comparing the complexity of algorithms used in various data structures.

Method of Collection: Each student is given a score on each of the three tests (40%), the final exam, (25%), and nine programming projects and homework assignments (35%).

Performance Levels	Frequency	Percentage
No Ability (F)	0	0%
Some Ability (D)	3	30%
Adequate Ability (C)	2	20%
More than Adequate Ability (B)	3	30%
High Ability (A)	2	20%

Observations:

1. The majority of the students had difficulty completing the programming assignments due to their lacking the basic knowledge and skills of OOP programming. So the instructor spent four weeks to review the OOP topics. In spite of the effort, three out of 10 or 30% of the students received D's for the course, which effectively meant F, as they must repeat the course. So grade control becomes an important issue that needs to be addressed.

2. Reviewed algebra (mainly on logarithms and algebraic series) to pave the way to introduce the topic on analysis of algorithms.
3. In the end, 50% of the students who completed the course achieved "More than Adequate Ability" or "High Ability"; in spite of the fact that 30% of class effectively failed the course.