

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: S6:

Demonstrate abilities to select appropriate data structures and to design algorithm to solve problems.

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

Curriculum Committee Subgroup: **Data Structures and Algorithms**

Members: John Najarian (Chair), Erh-Wen Hu, Gilbert Ndjatou

Date: November 21, 2017

Updated On: January 30, 2018

A. Analysis of the Assessment Data

For the Fall 2015 – Spring 2017 assessment period, this student outcome was assessed in the CS 3420 Data Structures course with 21 students in the Spring 2016, 19 students in the Fall 2016, and 15 students in the Spring 2017 semester for a total of 55 students. Among those students, only 4 of them (which represents less than 2%) performed at a level below adequate. .

B. Suggestions for Improvement

N/A

C. Improvement Implemented

N/A

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

- a. Faculty Course Assessment Report: CS3420, Spring 2016

Data Collected: Each student’s level of performance on the lab assignments and the questions on the tests and the final exam.

Method of Collection: lab assignments, questions on tests and the final exam that assess students ability to implement, manipulate and use linked list, stacks, queues, and trees to solve problems.

Performance Levels	Frequency	Percentage
No Ability (level of performance of F)	0	0
Some Ability (level of performance of D)	1	4.8
Adequate Ability (level of performance C)	8	38.1
More than Adequate Ability (level of performance of B)	8	38.1%
High Ability level of performance of A)	4	19%

Observation:

- b. Faculty Course Assessment Report : CS3420, Fall 2016

Data Collected: Each student’s level of performance on the lab assignments and the questions on the tests and the final exam.

Method of Collection: lab assignments, questions on tests and the final exam that assess students ability to implement, manipulate and use linked list, stacks, queues, and trees to solve problems.

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

Observations:

- Spent four weeks reviewing CS2400 CS II topics including arrays, pointers and dynamic memory in response to students' request.
 - Some students had trouble handing in earlier programming assignments in a timely manner. To address the problem, the instructor organized students into small discussion groups and spent extra after-class hours with the groups turn the situation around.
 - Most students work very hard as evidenced by the extra hours many had chosen to stay in the classrooms after the class was over. Most students hand in nearly all programming assignments.
 - The instructor occasionally invited some students to present their code in class. Students seemed to be more enthusiastic and willing to participate in the discussion. Most would tell me that it really helped them learn and request more such opportunities.
-

c. Faculty Course Assessment Report: CS3420, Spring 2017

Data Collected: Each student's level of performance on the lab assignments and the questions on the tests and the final exam.

Method of Collection: lab assignments, questions on tests and the final exam that assess students ability to implement, manipulate and use linked list, stacks, queues, and trees to solve problems.

Performance Levels	Frequency	Percentage
No Ability	0	0.0%
Some Ability	2	20.0%
Adequate Ability	1	33.3%
More than Adequate Ability	8	33.3%
High Ability	4	13.4%

Observations:

- Spent more than three weeks on reviewing all OOP topics taught in the previous course and on dynamic memory, pointers, and arrays (static and dynamic; one and two dimensions) as most students had trouble with these topics.
- Some students had trouble handing in earlier programming assignments in a timely manner. To address the problem, the instructor organized students into small discussion groups and spent extra after-class hours with the groups turn the situation around.
- Most students work very hard as evidenced by the extra hours many had chosen to stay in the classrooms after the class was over. Most students hand in nearly all 12 programming assignments.