William Paterson University of New Jersey
College of Science and Health
Department of Computer Science
Analysis of Course Coverage and Assessment Report Data
Fall 2018 – Spring 2019 Assessment Cycle

Course Number: CS 2600 (Discrete Structures)

Course Coordination Committee Members: Cyril S. Ku

Date: May 28, 2019

A. Course Pre-requisites/Co-Requisites

a) Problems/Issues Identified:

The instructors mentioned that the CIT students were less prepared than the CS students.

b) Suggestions for Improvement:

As a department, we need to discuss this situation. Perhaps need to design two separated courses for discrete mathematics – one of CS majors and one for CIT majors.

B. Course Objectives

a) Problems/Issues Identified:

Course objectives hard to meet with so much materials to cover. The instructors mentioned that there are too many topics to be covered and suggested to make this course a 4-credit course.

b) Suggestions for Improvement

4-credit course has many consequences on scheduling, curriculum and credit adjustment for the major. All the instructors of CS 2600 and any course instructors who teach courses that need CS 2600 as prerequisite should get together and discuss the precise topics that need to be covered before deciding whether 4-credit course is feasible and practical.

C. Course Student Outcomes

a) Problems/Issues Identified:

Some learning outcomes are hard to measure.

b) Suggestions for Improvement

As suggested above, when everyone who teaches CS 2600 gets together to discuss the topics, they need to define the common student learning outcomes and their measurements.

D. Course Content

c) Problems/Issues Identified:

No problem identified, basically the same comment about the amount of materials need to be covered.

d) Suggestions for Improvement

See similar suggestions under Course Objectives above.

E. Support for the Attainment of the CS Program Student Outcomes

Student Outcome S2:

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

CS2600 is a discrete math course in which we discuss topics that include elementary propositional and predicate logics; elementary set theory; relations and their properties; functions; congruence and Euclidean algorithm; combinatorics; mathematical reasoning; matrices; elements of graph theory; trees and their applications; and Boolean algebra. Two of its major objectives are to emphasize mathematical reasoning and to show the applications of discrete mathematics.

However, this student outcome will no longer be assessed in this course because instructors do not always have the time to emphasize and assess the applications of the concepts covered in the class to the discipline of computer science.

F. Analysis of the Results of the Evaluations of the Course Student Outcomes Assessment Data and Suggested Improvements

The results of the evaluations of the assessment data of the course learning outcomes are listed in Section G below. Both instructors did not teach Prolog, perhaps due to their comments that there were too much to cover in this course. The grades/scores of the satisfaction level were pretty consistent. Both sets of data suggested that students had difficulty in mathematical proofs. L6 (proofs) consisted of the lowest satisfaction percentage. The next set of lower scores were L5 (predicate logic). As suggested under Course Objectives Improvement above, our department needs to discuss the right topics to cover for CS major and/or CIT major for this course.

G. Results of the Evaluations of the Course Student Outcomes Assessment Data

Course: CS 2600 Discrete Structures

Instructor: Judith A. Coomes

Semester: Fall 2018

	Student Outcomes	Where Measured	Percentage of Satisfactory Results*
L1	Know the general concept of discrete mathematics and its applications.	Semester Grades	85%
L2	Learn propositional logic AND, OR NOT IF-THEN, IFF	Quiz, Test, Final Exam	62%
L3	Understand the various forms of conditionals and necessary and sufficient conditions	Quiz, Test , Final Exam	60%
L4	Learn about argument and its validity and the rules of inference	Quiz, Test, Final Exam	52%
L5	Learn about predicate logic and its quantifiers (universal and existential)	Quiz, Test,	42%

		Final Exam	
L6	Know how to do mathematical proofs (direct, counterexample, indirect, etc.	Quiz, Test, Final Exam	40%
L7	Learn about elementary concepts of number theory	Quiz, Test, Final Exam	62%
L8	Understand sequences, summation, and recursive definitions	Quiz, Test, Final Exam	52%
L9	Learn the basic set theory and Venn diagrams	Quiz, Test, Final Exam	60%
10	Understand combinatorics, (factorial, permutation, combination, probability)	Quiz, Test, Final Exam	79%
11	Learn the elementary concepts of graph theory.	Quiz, Test, Final Exam	85%
12	Understand functions and relations. .	Quiz, Test, Final Exam	50%
13	Gain a general understanding of algorithm complexity. (Big-O)	Quiz, Test, Final Exam	78%
14	Learn the applications of discrete structures such as trees, Boolean algebra , logic gates, etc.	Quiz, Test, Final Exam	70%
15	Write a simple Prolog program	N/A	N/A

*** Notes:**

1. For all the scores, the percentage corresponds to the number of students who receive a score of at least 70% on the question(s) related to the learning outcome.
2. Semester grade consists of composite scores of homework, projects, and exams. The grades were slightly curved.
3. There were 7 quizzes, 3 full period tests, and a comprehensive final examination. A series of (ungraded) 15 worksheets was used during class periods and for homework in addition to the text.
4. All percentages are based on those final exam problems that pertain to the learning outcome, or on the semester grades.
5. The biggest problems appear to be predicate logic, quantifiers and mathematical proofs. These topics obviously need more time and practice. Perhaps they should be done as early as possible in the semester so that they are not rushed for time. An additional quiz or hand-in assignment might be in order.

Course: CS 2600 Discrete Structures

Instructor: Weihua Liu

Semester: Spring 2019

1	Know the general concept of discrete mathematics and	Midterm Exam, Final	100%

	its applications.	Exam	
2	Learn propositional logic; construct truth table of logical forms for logical operations such as AND, OR, NOT, IF-THEN, IFF.	Quiz 1, Midterm Exam, Final Exam	66.7%
3	Understand the various forms of conditionals: contrapositive, converse, inverse, necessary and sufficient conditions.	Quiz1, Quiz 2, Midterm Exam, Final Exam	50%
4	Learn about argument and its validity, and the rule of inference.	Quiz2, Midterm Exam, Final Exam	50%
5	Learn about predicate logic and its quantifiers (universal and existential).	Midterm Exam, Final Exam	50%
6	Know how to do mathematical proofs (direct, counterexample, exhaustion, indirect, contradiction, contraposition, induction).	Quiz 2, Midterm Exam, Final Exam	45.8%
7	Learn about the elementary concepts of number theory.	Quiz 5, Midterm Exam, Final Exam	62.5%
8	Understand sequences, summation, and recursive definition and algorithms.	Quiz 5, Final Exam	66.7%
9	Learn the basic set theory and Venn diagram.	Quiz2, Midterm Exam, Final Exam	45.8%
10	Understand combinatorics (factorial, permutation, combination, probability).	Quiz5, Quiz 6, Final Exam	75%
11	Learn the elementary concepts of graph theory	Final exam	58.3%
12	Understand functions and relations.	Quiz 3,Midterm Exam, Final Exam	62.5%
13	Gain a general understanding of algorithm complexity (e.g., Big-O)	Midterm Exam, Final Exam	70.9%
14	Learn the applications of discrete structures such as trees, Boolean algebra and logic gates, etc	Quiz2, Midterm Exam, Final Exam	62.5%
15	Know how to write a very simple Prolog program	N/A	

*** Notes:**

6. For all the scores, the percentage corresponds to the number of students who receive a score of at least 70% on the question(s) related to the learning outcome.
7. Semester grade consists of composite scores of homework, projects, and exams. The grades were (curved/not curved).