

**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 3500 (Software Engineering)

**Course Coordination Committee Members:** Cyril S. Ku

**Date:** June 8, 2019

**A. Course Pre-requisites/Co-Requisites**

**a) Problems/Issues Identified:**

Object-oriented programming experience is very important to design a software project using UML. Students should have CS 2400 (Computer Science II) or CS 2420 (Object-oriented Programming with Java) experience. Since CS students need CS 3420 as pre-requisite for this course, they will have CS 2400 requirement. For CIT students, the current prerequisite is CS 2550. Need to add CS 2400/2420 as part of the prerequisite.

**b) Suggestions for Improvement:**

The CS Department has been re-examining all the course pre-requisites currently and will modify the pre-requisite structures of some courses in the coming semester.

**B. Course Objectives**

**a) Problems/Issues Identified:**

No problem identified.

**b) Suggestions for Improvement**

N/A

**C. Course Student Outcomes**

**a) Problems/Issues Identified:**

No problem identified.

**b) Suggestions for Improvement**

N/A

**D. Course Content**

**c) Problems/Issues Identified:**

No problem identified.

**d) Suggestions for Improvement**

N/A

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

### **1. Student Outcome S1: Communicate effectively in a variety of professional contexts.**

A requirement for this course is a team project for which students have to produce documentations about a software development life cycle. The instructor usually splits the documentation according to the different stages of requirements: specification, design, and implementation, and the design part of the project which is presented to the class is graded by students and the instructor of the course. The report itself is graded by the instructor. This course is therefore consistent with this student outcome.

### **2. Student Outcome S4:**

**Function effectively as a member or leader of a team engaged in activities appropriate to the program's discipline.**

A major objective of this course is to produce a group project. The instructor usually divides the class into several 3- to 4-person teams (dependent on enrollment). Each team needs to produce documentations which require substantial coordination among team members. Students need to meet outside class time to work on the project. Tests and final exam include questions of team organization and management. This course therefore supports the attainment of this student outcome.

### **3. Student Outcome S8:**

**Analyze a complex computing problem and to apply principles of computing and other relevant disciplines to identify solutions.**

One of the major objectives of this course is for the students to produce a team project. Students need to use software engineering principles, data structures, and object-oriented programming paradigm to analyze and design large software projects. The instructor usually splits the documentation according to the different stages of requirements: specification, design, and implementation, and the design part of the project which is presented to the class is graded by students and the instructor of the course. The report itself is graded by the instructor. This course is therefore consistent with this student outcome.

## **F. Analysis of the Results of the Evaluations of the Course Student Outcomes Assessment Data**

The assessment data of the course learning outcomes are listed in Section G below. All the sections were taught by the same instructor and the course materials were relatively the same. The disparity in scores/grades for these different sections were mostly due to the quality of the students. Put object-orientation requirement (prerequisite) for this course should improve the performance of students. The N/A satisfactions were due to the fact that the outcome was covered but not tested. The obvious improvement can be made is to have a question or two to cover the legal and ethical behaviors in software engineering. Exam #2 was where the object-oriented concepts were tested. The scores were low and the above suggestion of prerequisite should help remedy this issue. The final exam scores

were relatively low due to the comprehensive nature of the exam. The instructor will consider to lower the percentage of the final exam to distribute more evenly the percentages for all the exams.

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

Course: CS 3500-60 (Software Engineering)  
 Instructor: Cyril S. Ku  
 Semester: Fall 2018

	Student Outcomes	Where Measured	Percentage of Satisfactory Results*
L1	Understand the phases of software lifecycle of requirements, design, implementation, testing, installation, and operation and maintenance.	Exam #1	60%
L2	Explain software development process and object-oriented methodology.	Exam #2	32%
L3	Perform analysis of a problem to determine user needs and develop specifications of the system required that will meet user needs within economic constraints.	Project #1	60%
L4	Perform design from specification and generate codes from design using UML.	Project #2	100%
		Project #3	68%
		Final Exam	32%
L5	Identify approaches and standards for software quality improvement.	All Projects	48%
		Final Exam	32%
L6	Gain general concepts of software project management.	Exam #3	72%
		All Projects	48%
L7	Understand the legal and ethical behaviors in the software industry, as well as the professional code of conduct of a software engineer.	N/A*	N/A*
L8	Produce a project documentation that consists of four major parts of the software development process: requirements, specification, design, and implementation.	All Projects	48%

**\*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. However, the final grades had higher level of satisfactory percentage because they were curved.
2. Semester grade = Projects (35%) + Exam #1 (15%) + Exam #2 (15%) + Exam #3 (15%) + Final Exam (20%)
3. L7 was discussed and covered in lecture but it was not tested. However, students practiced ethical conduct in their projects.

Course: CS 3500-01 (Software Engineering)  
 Instructor: Cyril S. Ku  
 Semester: Spring 2019

	Student Outcomes	Where Measured	Percentage of Satisfactory Results*
L1	Understand the phases of software lifecycle of requirements, design, implementation, testing, installation, and operation and maintenance.	Exam #1	42%

L2	Explain software development process and object-oriented methodology.	Exam #2	31%
L3	Perform analysis of a problem to determine user needs and develop specifications of the system required that will meet user needs within economic constraints.	Project #1	88%
L4	Perform design from specification and generate codes from design using UML.	Project #2	100%
		Project #3	88%
		Final Exam	38%
L5	Identify approaches and standards for software quality improvement.	All Projects	88%
		Final Exam	38%
L6	Gain general concepts of software project management.	Exam #3	73%
		All Projects	88%
L7	Understand the legal and ethical behaviors in the software industry, as well as the professional code of conduct of a software engineer.	N/A*	N/A*
L8	Produce a project documentation that consists of four major parts of the software development process: requirements, specification, design, and implementation.	All Projects	88%

**\*Notes:**

- 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. However, the final grades had higher level of satisfactory percentage because they were curved.
- Semester grade = Projects (35%) + Exam #1 (15%) + Exam #2 (15%) + Exam #3 (15%) + Final Exam (20%)
- L7 was discussed and covered in lecture but it was not tested. However, students practiced ethical conduct in their projects.

Course: \_\_\_\_\_ CS 3500-60 (Software Engineering)  
 Instructor: \_\_\_\_\_ Cyril S. Ku  
 Semester: \_\_\_\_\_ Spring 2019

	Student Outcomes	Where Measured	Percentage of Satisfactory Results*
L1	Understand the phases of software lifecycle of requirements, design, implementation, testing, installation, and operation and maintenance.	Exam #1	67%
L2	Explain software development process and object-oriented methodology.	Exam #2	38%
L3	Perform analysis of a problem to determine user needs and develop specifications of the system required that will meet user needs within economic constraints.	Project #1	88%
L4	Perform design from specification and generate codes from design using UML.	Project #2	100%
		Project #3	63%
		Final Exam	50%
L5	Identify approaches and standards for software quality improvement.	All Projects	63%
		Final Exam	50%
L6	Gain general concepts of software project management.	Exam #3	71%
		All Projects	63%
L7	Understand the legal and ethical behaviors in the software industry, as well as the professional code of conduct of a software engineer.	N/A*	N/A*

L8	Produce a project documentation that consists of four major parts of the software development process: requirements, specification, design, and implementation.	All Projects	63%
----	---	--------------	-----

**\*Notes:**

7. 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. However, the final grades had higher level of satisfactory percentage because they were curved.
8. Semester grade = Projects (35%) + Exam #1 (15%) + Exam #2 (15%) + Exam #3 (15%) + Final Exam (20%)
9. L7 was discussed and covered in lecture but it was not tested. However, students practiced ethical conduct in their projects.