Spring 2022 CQI Report
Software Curriculum

September 22, 2022; revised on October 10, 2022

Committee: Yoonsik Cheon, Bhanukiran Gurijala, Daniel Mejia, Oscar Mondragon, and Salamah Salamah.

The Software cluster consists of four courses: CS 3195 Junior Professional Orientation, CS 3331 Advanced Object-Oriented Programming, CS 4310 Software Engineering I, and CS 4311 Software Engineering II. The committee and the course instructors met on September 20, 2022, and reviewed the assessments of the learning outcomes provided by the course instructors; see the appendix for the individual assessment reports prepared by the instructors.

- CS 3195 by Daniel Mejia
- CS 3331 by Daniel Mejia
- CS 4310 by Oscar Mondragon
- CS 4311 by Yoonsik Cheon

The stated learning outcomes of all the courses were met except for three (two from CS 4310 and one from CS 4311). However, several outcomes were met marginally with the average earned points of 70% - 74%.

Revision of Course Outcomes (Need Faculty Approval)
CS 3195 and CS 4310 instructors proposed several changes to the current course outcomes, and the committee reviewed/discuss them carefully. Below are the proposed changes that the committee approved. All the proposed changes except for CS 3331 Outcome 1a were approved by faculty at the faculty meeting on Friday, October 7.

CS 3195: Junior Professional Orientation

- Update outcome 1a: Describe techniques for face-to-face and telephone interviews.
    - Describe techniques for face-to-face, *virtual,* and telephone interviews
- Update outcome 2b: Prepare for and participate in a mockup interview.
    - Prepare for and participate in a mockup interview *to improve professional or technical oral communication*
- Update/Split outcome 2d: Prepare a portfolio that includes a cover letter, resume, samples of software development experiences, oral communication, and written communication samples to be the following:
    - Prepare a professional networking account (e.g., LinkedIn) to establish networking and professional visibility for the purpose of enhancing written communication skills, obtaining internships or research experiences, obtaining graduate program information, and obtaining full-time positions.
    - Develop a professional resume/cv to express professional experiences.
    - Create a professional repository (e.g., Github) to showcase software development experiences.

CS 3331: Advanced Object-Oriented Programming

- Remove Level 1a outcome: Explain the differences between an object-oriented approach and procedural approach.
  - Move to CS 3360 Programming Languages.
- Add a new Level 1 outcome:
  - Describe the difference between waterfall and agile software development.
- Rewrite 3b into multiple outcomes: Design, implement, and use classes and methods that follow conventions and styles, and make appropriate use of advanced features such as inheritance, exception handling, and generics.
  - Design, implement, and use classes and objects by following coding conventions, guidelines, styles, and standards.
  - Design and implement exception handling (including user defined exceptions) and high order functions.

CS 3195 Junior Professional Orientation
This course provides an overview to the Computer Science profession with an emphasis on ethics and the local and global impact of computing on individuals, organizations, and society. In spring 2022, it was taught by Daniel Mejia using the textbook, *A Gift of Fire: Social, Legal, and Ethical Issues for Computing Technology* (Sara Basse, 5th edition, Pearson, 2018). The committee recommends to:

- Keep up with the good work – meeting all the course outcomes satisfactory with a 100% student passing rate.
- Expand on the use of technology for communication and collaboration.
- Continue having student support in the form of an Instructional Assistant (IA).

CS 3331 Advanced Object-Oriented Programming
This course provides in-depth exposure to the object-oriented programming paradigm. In spring 2022, it was taught by Daniel Mejia. The required textbook was *Object-Oriented Analysis, Design, and Implementation: An Integrated Approach* (Brahma Dathan, Sarnath Ramnath. Springer, Universities Press, 2015). The committee recommends to:

- Keep up with the good work meeting all the course outcomes.
- Continue monitoring the failure rate (17%); it was 24% in the previous round.
- Consider using a project-based learning approach in this course (i.e., a semester long project that builds upon this course)
- Use a software development approach such as waterfall or agile for the course project.

CS 4310 Software Engineering I
This course is the first semester of a two-semester capstone course in which students work with a customer to capture and specify requirements for a real-world application. In spring 2022, it was taught by Oscar Mondragon. The course textbook was *Requirements Engineering* (Hull, E., Jackson, K., and Dick, J., 4th edition, Springer, 2017). The committee recommends to:

- Monitor the two outcomes that were not met (1a. Software Engineering principles) and 2e. application of the code of ethics).
- Suggest to the administration to have a class capacity of 50 students per section and a pool of instructors to ensure the course quality.

CS 4311 Software Engineering II

This course is the second semester of a two-semester capstone course in which students design and implement a real-world application specified in CS 4310. In spring 2022, it was taught by Yoonsik Cheon. The required textbooks were *Essential of Software Engineering* (F. Tsui, O. Karam, and B. Bernal, 3rd Edition, Jones & Bartett Learning, 2014) and *Designing Object-Oriented Software* (R. Wirfs-Brock, R. Wilkerson, and L. Wiener, Prentice Hall, 1990). The committee recommends to:

- Monitor the outcomes that were not met (2b. detailed design) or met marginally (2a. architectural design and 2j. analysis of non-functional properties).
- The course has two required textbooks: one for theory and the other for practice (CRC approach). Consider updating the old CRC textbook or creating supplementary materials, e.g., a CRC handbook or workbook.
- Let the students use the UML notation to document their CRC-based design work; nobody knows or uses the old CRC notation today. And provide more examples.
- There is a process mismatch between what students are taught for their project design work (waterfall lifecycle model) and what they do for the actual implementation (incremental and iterative approach). Consider streamlining the two processes, e.g., incorporating the CRC technique into a modern development approach such as Agile development.
- Work with the CS 3331 and 4310 instructors to strengthen the (individual) practice of UML class diagrams.

Appendix

CS 3195-cqi-spr2022.docx (Daniel Mejia)
CS 3331-cqi-spr2022.docx (Daniel Mejia)
CS 4310-cqi-spr2022.docx (Oscar Mondragon)
CS 4311-cqi-spr2022.docx (Yoonsik Cheon).