Program Assessment Report (PAR)

Computer Science (CS) Masters Degree

REPORT PREPARED by: Pei, Yong

ACADEMIC YEAR COVERED BY THIS REPORT: 2021-2022

I. PROGRAM LEARNING OUTCOMES

The program balances theory, software, hardware, and practice with coursework-only and thesis options available. It offers a wide range of courses in computer science and the opportunity to develop research skills in computer science areas. The program's strengths include diverse faculty expertise, various computer science laboratories, and a balance of theory and practice. Degree requirements focus on the areas of software systems design and analysis, and computer science theory. Program Learning Outcomes Graduates of the Master's of Science program in Computer Science will be able to • apply abstract reasoning to complex problems in computer science; and • solve a broad range of problems through the application of current computer science techniques. The program learning outcomes are further mapped into the following specific course learning outcomes • Ability to design algorithms using dynamic programming, greedy algorithms, and divide and conquer approaches • Develop interpreter-based specification (operational semantics) of programming languages • Transform formal specifications to design and code • Should be able to write medium sized programs using distributed computing languages and/or libraries

II. PROCEDURES USED FOR ASSESSMENT

A. Direct Assessment

(i) Assessment Schedule Completed Program Learning Outcome Data Collection Term Review Term 1. Abstract reasoning Annual Fall 2020, Fall 2021 2. Problem solving Annual Fall 2020, Fall 2021 (ii) Alignment of program learning outcome to course learning outcomes 1. Abstract reasoning CS 7200 Ability to design algorithms using dynamic programming, greedy algorithms, and divide and conquer approaches 2. Problem solving CS 7100 Develop interpreter-based specification (operational semantics) of programming languages CS 7140 Transform formal specifications to design and code CEG 7370 Should be able to write medium sized programs using distributed computing languages and/or libraries (iii) Course learning outcomes

to assignment being collected and assessed CS 7200 Ability to design algorithms using dynamic programming, greedy algorithms, and divide and conquer approaches Final exam questions CS 7100 Develop interpreter-based specification (operational semantics) of programming languages Programming projects CS 7140 Transform formal specifications to design and code Programming projects CEG 7370 Should be able to write medium sized programs using distributed computing languages and/or libraries Programming projects (iv) Collection of student artifacts assessed Fall 2018 CS 7200 Final exam questions, grade distributions, student work. Fall 2018 CS 7100 Programming project descriptions, grade distributions. Fall 2020 CS 7140 Programming project descriptions and student work. Fall 2020 CEG 7370 Programming project descriptions and student work.

B. Scoring of Student Work

The program learning outcomes are mapped to specific learning outcomes of the program core courses. Core courses are those that are required as a part of each student's Program of Study (POS). Therefore, each program outcomes are measured at least once over the course of a student's POS. Learning outcomes are directly assessed by evaluating the student performance in corresponding project assignments and exams.

C. Indirect Assessment

The program educational outcomes were established with input from and review by the external advisory board (EAB). In addition, the advisory board has reviewed and expressed approval of all major program changes made in the last five years. The Department of Computer Science and Engineering external advisory board includes representatives of local, regional and other businesses that historically hire Department graduates, as well as successful alumni of our programs. The board meets each Fall and Spring semester to review program objectives, curriculum and program changes, and new programs and courses. They make both high-level strategic recommendations and specific course and curriculum suggestions to the program. "College of Engineering and Computer Science, Master of Science Assessment of Learning Outcomes During Exit Interview" surveys are used as additional measures for indirect assessment. Survey is instrumented to collect graduating student assessment of self-efficacy for the learning outcomes. For each outcome students are asked to rate their own level of ability/achievement.

III. ASSESSMENT RESULTS/INFORMATION:

CS 7200 - Final exam questions. Grade distributions. Student work. CS 7100 - Programming project descriptions, grade distributions. CS 7140 - Programming project descriptions, grade distributions, student work. CEG 7370 - Programming

project descriptions and student work. Grade distributions.

Course learning outcome achieved. No concern is raised. Course learning outcome achieved. No concern is raised. Course learning outcome achieved. No concern is raised. Course learning outcome achieved. No concern is raised.

Ability to design algorithms using dynamic programming, greedy algorithms, and divide and conquer approaches Develop interpreter-based specification (operational semantics) of programming languages Transform formal specifications to design and code Should be able to write medium sized programs using distributed computing languages and/or libraries

IV. ACTIONS TO IMPROVE STUDENT LEARNING

The program institutes a formal assessment program involving the collection of students performance checkpoint data related to each educational outcome of the program. Data have been collected since Fall, 2018. A formal program assessment was conducted by the Graduate Studies Committee of the Department of Computer Science and Engineering in Fall 2020 and Fall 2021. This data and assessment results are shared through Pilot among the GSC faculty members and student advisors. The GSC faculty and student advisors review all program courses every three years to ensure that course pre-requisites are relevant, student learning objectives are accurate and sequential courses are aligned. In response to the continual program assessments, in the last five years the Graduate Studies Committee has enacted several significant changes to the program curriculum designed to increase flexibility and student retention, while maintaining program rigor. Among these changes 1) The number of core courses was reduced from three to two, with every student required to take the core course CS 7200 - Algorithm Design and Analysis. 2) In recognition of increasing levels of specialization in the field, the number of 6000-level courses allowed in the program was increased from two to four for non-thesis students, allowing students to take both introductory and advanced courses in more of their areas of interest. 3) A new course, CS 5160 - Fundamentals of Computer Science, was created and offered to allow students from non-CS technical majors an onramp into the graduate program. This course has been highly successful, bringing in many new successful students that would previously have been unable to join or succeed in the program. 4) New courses are created as needed to ensure coverage of emerging technologies.

V. SUPPORTING DOCUMENTS

Additional documentation, when provided, is stored in the internal Academic Program Assessment of Student Learning SharePoint site.