NEW YORK INSTITUTE OF TECHNOLOGY

Continuous Program Improvement (CPI)

Student Learning Outcomes (SLO)/Program Learning Outcomes (PLO)

Plan Implementation Report - AY 2023-24

Program name	MS Computer Science
Expected date of submission	6/30/2024
Department chair/program director	Steven Billis
Dean's signature	

New York Tech's CPI process is implemented to meet Middle States Commission on Higher Education (MSCHE) Standard V: *Educational Effectiveness Assessment*, which states: "Assessment of student learning and achievement demonstrates that the institution's students have accomplished educational goals consistent with their program of study, degree level, the institution's mission, and appropriate expectations for institutions of higher education."

Each department was asked to create a three-year assessment/evaluation plan to improve student learning for ***each of their degree programs*** covering the following academic years: **2022-2023, 2023-2024, and 2024-2025.**

All degree programs' three-year Program Learning Outcomes (PLO) plans are available here: http://www.nyit.edu/planning/academic_assessment_plans_reports

This is a report on the PLO CPI plan **implementation** for the **2023-24** academic year.

First, please respond to the feedback provided by the CPI Committee in response to your program's prior year (AY 2022-23) CPI plan implementation report. How did you incorporate the Committee's recommendations into your CPI efforts?

Second, please address the following points in this year's (AY 2023-24) report:

1. Program learning outcomes assessed
   List the program learning outcomes that were assessed in AY 2023-24 based on your three-year plan (2022-25).
   (Please refer to the guidelines for articulating expected program learning outcomes.)

2. Methods
   Describe the method of assessment that you used (student artifacts, sampling methods, sample size, who and how they were assessed, etc.) and attach measurement instruments (e.g., rubrics, exam items, scoring guide for a particular task, supervisor evaluation form, survey instrument, and other measurement tools). Remember: direct assessment is required, and both direct and indirect assessment are strongly recommended.
   (Please refer to the guidelines for assessment methods.)

3. Analyze and interpret assessment data
   It is strongly recommended to provide criteria-based analyses of assessment results and based on the analysis to determine if students are meeting the expected learning outcomes.
   (Please refer to the guidelines for compiling, analyzing and interpreting assessment data).

4. Close the Loop
   If the expected program learning outcomes were successfully met, describe how the program will keep or expand the good practices. If they were not successful, explain how you have or will refine the plan and begin the next cycle of Plan-Do-Study-Act (PDSA).
   (Please refer to the guidelines for closing the loop and taking action to improve program learning outcomes.)

5. Describe how faculty were involved in the implementation of the PLO CPI plan and how the results will be communicated to all stakeholders.

The Student Learning Outcomes (or Program Learning Outcomes) of the MS in CS are:

1. A comprehensive background in theory and design of assemblers, compilers, and operating systems.
2. A comprehensive knowledge of computer architecture.
3. A comprehensive knowledge of mathematical & algorithmic concepts & analysis
4. Proficiency in specific areas of specialization such as computer security, software engineering, computer graphics, and artificial intelligence.
5. A comprehensive knowledge of analysis, design, and development of a computerized system.

The matrix relating PLOs and the graduate CS courses we will be using is given below:

Course	PLO1	PLO2	PLO3	PLO4	PLO5
CSCI 610 Theoretical Concepts in Computers and Computation			●	●	
CSCI 620 Operating System Security			●		
CSCI 621Programming Languages	●	●			●
CSCI 641 Comp. Architecture		●			
CSCI 665 Software Eng.				●	

Timeline for the PLO Assessment

Program Learning Outcomes	AY 22-23	AY 23-24	AY 24-25
1. A comprehensive background in theory and design of assemblers, compilers, and operating systems.	●		
2. A comprehensive knowledge of computer architecture	●		
3. A comprehensive knowledge of mathematical & algorithmic concepts & analysis		●	
4. Proficiency in specific areas of specialization such as computer security, software engineering, computer graphics, and artificial intelligence.		●	
5. A comprehensive knowledge of analysis, design, and development of a computerized system.			●

To set the context for the program's assessment activities it is useful to understand the role of this work within the larger institution. New York Tech implemented Continuous Program Improvement (CPI) in 2020 across all academic departments and students support units to improve educational effectiveness. It replaced the Academic Assessment Committee of the Senate. CPI emphasizes a data-informed, decision-making process to guide departments for overall quality improvement that leads to the improvement of students' learning, college experiences, and achievement.

The CPI Committee of the Academic Senate is the institutional unit that brings together all assessment and improvement activities at the university—for programs with or without professional accreditation, and for academic departments and student support units. The committee members come from all academic schools and numerous support departments. Its meetings are open, and minutes are posted on the web site of the Academic Senate.

The Committee's mission is to:

- Raise the visibility of CPI for educational effectiveness assessment within the university
- Maintain a common, unified, mission-driven process

- Improve educational effectiveness by increasing faculty participation in and knowledge of science of improvement
- Prepare a formal annual report on the status of assessment at the university, including recommendations for improvement
- Ensure that the Continuous Program Improvement (CPI) process is used to advance New York Tech's mission and goals and connected with financial planning and support
- Periodically evaluate (CPI) process and make recommendations for improvements

NYIT's model for the assessment of student learning in its academic programs is designed according to the following principles:

- Program faculty are responsible for assessing the student learning outcomes of their program.
- Assessment activities should be useful, annual, and integrated as much as possible into what faculty are already doing.
- Faculty define the most important learning outcomes, set standards of performance, and measure achievement.
- Results are used to make program improvements.
- The CPI Committee of the Academic Senate provides institutional oversight.
- The offices of the Provost and the Vice President for Research, Assessment and Decision Support provide institutional support.

At NYIT’s College of Engineering and Computing Sciences, each program has a multidimensional assessment process in place to ensure that the Student Outcomes have been attained. It is a process that provides data to support continuous program improvement.

To ensure that students achieve student outcomes 1 to 5, the faculty has built the curriculum such that key concepts are introduced, developed, and reinforced throughout students' time in the program.

In both fall and spring semesters, CS faculty members prepare a Faculty Course Assessment Report (FCAR) for each course they teach. The FCAR requires:

- The FT faculty members of the department have met previously, as a group, to determine the relationship between the SOs and the CS program's required and elective courses.
- The FT or adjunct faculty teaching a specific course is required to establish appropriate performance tasks (APTs) to assess to what extent each SO is being met. These APTs may be quizzes, exam questions, reports, projects, presentations, etc.
- Each student's APT is then scored with the method shown below to create an EGMU vector for each SO and a corresponding assessment metric. It should be noted that the faculty member is required to show which part of each APT is being used to form a metric for the student outcome with appropriate documentation.

EGMU		Score
E - Excellent	Fully demonstrates/accomplishes the attributes and behavior in the rubric	3
G – Good	Mostly demonstrates/accomplishes the attributes and behavior in the rubric	2
M – Minimal	Minimally demonstrates/accomplishes the attributes and behavior in the rubric	1
U - Unsatisfactory	Does not demonstrate/accomplish the attributes and behavior in the rubric	0

These course-embedded assessments serve as the primary tools to determine student outcome achievement and afford a direct link between learning outcomes and student outcomes as one aspect of curriculum change.

The data from FCARs are then evaluated at the spring Faculty Assessment meetings. At these meetings all full-time faculty members and those regular part-time faculty members wishing to participate identify and propose strategies to improve Student Outcomes.

While many courses may satisfy a particular outcome, the assessment committee has picked a subset of these courses that it finds most appropriate to determine the minimum metric for each outcome.

The recommendations of the assessment committee meetings are generally of two types: One set of recommendations can be implemented solely through the faculty member making internal changes to the courses (i.e., textbook changes, pedagogical changes). The other set of recommendations would need to be forwarded to the curriculum committees of the College of Engineering and Computing Sciences and then to the Academic Senate for adoption (i.e., new course, prerequisite/co-requisite changes, catalog description).

We have found that each of our assessment tools must be used in conjunction with one another if we are to undertake changes that are meaningful.

AY 22-23

PLO2: To assess this PLO the department chose:

CSCI 641 Computer Architecture:

- Students were introduced to performance evaluation techniques and learned how to use the results of such techniques in the design of computing systems **EGMU 2.55**

PLO1: To assess this PLO the department chose:

CSCI 621 Programming Languages

- Students were tested on control structures for imperative, functional and logical programming languages **EGMU 2.35**

AY 23-24

PLO3: To assess this PLO the department chose:

CSCI 610 Theoretical Concepts in Computers and Computation

- Students were tested on graph theory, and combinatorics. **EGMU 2.65.**

PLO4: To assess this PLO the department chose:

CSCI 620 Operating System Security

- Students were tested on file system security and operating system integrity verification techniques. **EGMU 2.25**

AY 24-25

PLO5: To assess this PLO the department chose:

CSCI 621 Programming Languages

- Students were tested on the general principles of modern programming language design. **EGMU 2.35**

Closing the Loop

The faculty found that they could improve these EGMUs by spending more time on:

CSCI 641 superscalar with in-order and out of order execution
CSCI 610 spending more time on graph theory
CSCI 620 spending more time on cryptography

All CS full-time and adjunct faculty were involved in this exercise and the resulting report will be made available to our stakeholders (industrial board members (at IAB meetings), students and administration).