

University of Puerto Rico at Bayamon

Student Outcomes Update and Realignment of Performance Indicators Computer Science Program

Dr. Juan M. Solá Sloan

Assessment and Accreditation Coordinator

Last revision: September 2024

Document's Rationale

In March 2021, the document *Student Outcomes and Performance Indicators Revision* was prepared. It presents the student outcomes (a) to (j) of the Computer Science program with their corresponding performance indicators. The AAC was under the impression that mappings could be used to comply with Criterion 3 of the *Criteria for Accrediting Computing Programs* of the *Computing Accreditation Commission* of the *Accreditation Board of Engineering and Technology*. There was already a mapping between student outcomes (a) to (j) and (1) to (6) and it was presented in the document prepared in March 2021. However, in 2024 we had to adopt the student outcomes from (1) to (6) *at verbatim*. This event forced us to review/revise the alignment we made in 2021. This alignment is presented in this document. Note that the labels used for the performance indicators have not changed but they are directly related to the student outcomes adopted.

Student Outcomes and Performance Indicators Correspondence

The following list presents each SO with their corresponding performance indicators:

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

(a.1) Select the appropriate algorithm for a specific situation (Cog-Knowledge).

(a.2) Analyze the asymptotic running time of algorithms using big-O notation (Cog-Analysis)

(a.3) Apply mathematical concepts in the solution of a given problem (Cog-Application)

(b.1) Analyze a problem (Cog-Analysis)

2. Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline.

(b.2) Identify and define the computational requirements needed in a real situation (Cog-Synthesis).

(b.3) Choose the appropriate software and/or hardware tools to meet the desired goals (Cog-Evaluation).

(c.1) Design solutions using pseudo code, diagrams or natural language (Cog-Synthesis).

(c.2) Implement an algorithm using the appropriate programming language (Cog-Application).

(c.3) Perform both unit and systems testing (Cog-Evaluation).

3. Communicate effectively in a variety of professional contexts.

(f.1) Present different topics both orally and/or in writing (Affective-Responding).

(f.2) Explain technical concepts using the correct terminology (Affective-Valuing).

(f.3) Display knowledge of technical report writing skills (Cog-Knowledge)

4. Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles.

(e.1) Evaluate the ethical implications of an issue in the computing discipline (Cog-Evaluation).

(e.3) Recognize the responsibilities inherent to the profession (Cog-Knowledge).

(e.4) Evaluate the consequences of breaking the law (Cog-Evaluation).

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

(d.1) Evaluate a given problem within a team environment (Cog-Evaluation).

(d.2) Perform the duties assigned when working on a team (Affective-Responding).

6. Apply computer science theory and software development fundamentals to produce computing-based solutions.

(j.1) Determine the most appropriate data structures needed to solve a given problem.

(j.2) Appraise whether a given algorithm performed better than another.

(j.3) Determine the most appropriate programming paradigm needed to solve a problem

(j.4) Perform object oriented and structured analysis and design of software systems.

(j.5) Construct software systems of varying complexity.

Final Comments

After this change, it is imperative to update the already prepared document called *Final CIP Report*. The AAC foresees that the results obtained are **not** very far from those already obtained.