

Program Outcomes and Performance Indicators Revision

Our Program Outcomes did not undergo a significant change. However, we changed the way they were labeled. Previously we use numbers to label each one of them. We adopted the labels used by ABET for Program Outcomes since it was difficult to match our labeling system to ABET's. Our performance indicators were adjusted accordingly.

We reduced the number of performance indicators from 37 to 28. This was done by carefully analyzing our performance indicators and the way they supported the outcomes. Some of the performance indicators were eliminated while others were re-focus and expanded. Our performance indicators were labeled using letters and numbers, while the outcomes were labeled with letters instead of numbers.

The following list presents each outcome with their corresponding performance indicators.

a. An ability to apply knowledge of computing and mathematics appropriate to the discipline.

- (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)

This outcome suffered no change.

b. An ability to analyze a problem, identify and define the computing requirements appropriate to its solution.

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

This outcome suffered no change.

c. An ability to design, implement, and evaluate a computer-based system, process, component or program to meet desired needs.

- (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).

The performance indicators changed for this outcome. The previous performance indicators for this outcome (a) Design a solution for a given problem using the structured approach, (b) Design a solution for a given problem using the object-oriented approach, and (d) Implement abstract solutions using pseudo-code, flowchart or natural language, were consolidated in performance indicator (c.1). The AAC decided that this performance indicator was sufficient since students would use flowcharts, UML, Problem Analysis Charts or state diagrams to design a solution. All of them are valid measures if the student can abstract the problem and design a solution. Also pseudo-code and natural language are covered in (c.1). However, (c.2) and (c.3) are performance indicators that remain untouched from the previous cycle.

d. An ability to function effectively on teams to accomplish a common goal.

(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).

The performance indicators for outcome d changed. The previous performance indicators: (b) Performs the tasks assigned when working on a team, and (c) Assists its team mates when needed were consolidate into a modified version of (d) Complete its duties assigned within a team environment. The AAC concluded that outcome (d.2) encompasses the essence of these previously used performance indicators.

e. An understanding of professional, ethical, legal, security and social issues and responsibilities.

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

(e.2) Evaluate the social impact of a given computing technology (Cog-Evaluation).

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

This outcome suffered no change.

f. An ability to communicate effectively with a range of audiences.

(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)

This outcome suffered no change.

g. An ability to analyze the local and global impact of computing on individuals, organizations, and society.

(g.1) Understand computational or technological advances and their impact on individuals, organizations and society. (Cog-Comprehension).

(g.2) Recognize the global and local impact of a given technology (Cog-Knowledge).

(g.3) Be aware of the state of the art in computing technology (Cog-Comprehension).

Performance indicators for outcome g changed substantially. Previous performance indicators were: (a) Identify the contribution of computing and other related professionals to society, and (b) Understand computational or technological advances and their impact to the profession. Performance indicator (b) now is (g.1). The AAC analyzed this outcome and drafted (g.2) and (g.3). Previous performance indicator (a) was eliminated since the AAC understands that it was not supporting outcome g considerably as it should.

h. Recognition of the need for and an ability to engage in continuing professional development.

Performance indicators for outcome h were eliminated. Previous performance indicators were: (a) Recognize options of continuing studies after degree completion, (b) Use diverse information resources when performing assigned duties. The AAC determined that performance indicator (b) as written had no place under this outcome, leaving performance indicator (a) by itself. The AAC understands no need to outline the indicator (a).

i. An ability to use current techniques, skills, and tools necessary for computing practices.

(i.1) Use hardware and software tools currently available (Cog-Application).

(i.2) Use current techniques and skills in the practice of the profession (Cog-Application).

This outcome suffered no change.

j. An ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the trade-offs involved in design choices.

(j.1) Solve problems using the principles from discrete mathematics (Cog-Application).

(j.2) Solve problems using the principles from continuous mathematics (Cog-Application).

(j.3) Determine the most appropriate data structures needed to solve a given problem (Cog-Evaluation)

(j.4) Appraise whether a given problem has a computational solution (Cog-Evaluation).

(j.5) Determine the most appropriate programming paradigm needed to solve a problem (Cog-Evaluation).

Performance indicators for outcome j changed substantially. The first performance indicator that changed substantially was the first one. This was: (a) Solve problems using the principles from discrete and continuous mathematics. The AAC decided to expand this performance indicator in two (j.1) and (j.2). Now, the AAC can determine in what area of mathematics our program need to be strengthen. The next performance indicator that was changed was (b) Perform basic algorithmic analysis using big-O notation. This performance indicator was eliminated from outcome (j), since; it is measured also in outcome (a). Performance indicators (j.3), (j.4) and (j.5) remain untouched¹. Previous performance indicator, (d) demonstrates basic knowledge of scientific computing using numerical analysis, was eliminated. The AAC determine that this performance indicator is covered if the student attained (j.2).

k. An ability to apply design and development principles in the construction of software systems of varying complexity.

(k.1) Perform object oriented and structured analysis and design of software systems (Cog-Application).

(k.2) Construct software systems of varying complexity (Cog-Synthesis).

There were some performance indicators eliminated from this outcome. However, performance indicator (k.1) and (k.2) remain. The AAC determine that previous performance indicators: (a) Determine the feasibility of a proposed software systems, and (c) analyze and evaluate alternatives for acquiring or developing a software system was already covered by performance indicator (k.2).

¹ This were (c),(e) and (f)