

QuickStart Guide:
10 Steps to Creating a Comprehensive Curriculum Assessment Plan (CCAP)

This is an abbreviated guide for creating and submitting CCAPs as part of the assessment process in your department/program. Authors are encouraged to consult the complete CCAP Operations manual as this reference sheet provides only a brief overview. A completed CCAP must be submitted for each undergraduate and graduate degree program.

The following is a list of recommended steps for completing a CCAP.

1. *Create program objectives.* Department faculty should meet to discuss the program mission and, by extension, the objectives for the program. Brainstorming may take place at a formal or informal meeting; notes from this meeting can be used to develop themes that form the program's objectives. These objectives should capture the accomplishments of program graduates within 6-12 months of graduation.
2. *Create student learning outcomes.* Based on the objectives of the program, outcomes are created to capture the knowledge/skills that should be mastered by program completion. Each outcome should describe how a student will be different as a result of their learning experience; when possible, these outcomes should be industry specific.
NOTE: Objectives and outcomes should be measurable/observable. Refer to the Revised Bloom's Taxonomy for a list of appropriate verbs.
3. *Identify Existing Data Sources.* Once objectives and outcomes have been established, data sources need to be identified. These sources are student work products, also referred to as assessment artifacts, used to demonstrate levels of student proficiency in achieving one or more student learning outcomes. Examples include sample student papers, alumni survey data and student exit surveys. A list of possible data sources and be found in the Operation Manual.
4. *Standardize Measurement with Rubrics.* A rubric is a method for organizing and applying criteria to assessment student work products. This scoring tool is used to clearly outline performance expectations. These assessment rubrics will likely differ from grading rubrics in that they are solely based on student proficiency and don't consider other criteria such as tardiness, participation, etc.
5. *Establish Performance Criteria.* A threshold should be established that signifies an appropriate level of student learning. Some programs may have standards set by agency requirements; other may discuss with faculty and establish their own metric to support an appropriate achievement of student learning.
6. *Submit a CCAP.* Completed CCAPs can be submitted to the Office of Academic Assessment for review. Feedback will be provided to the individual submitting. Once a CCAP is accepted, it will be maintained by the Office of Academic Assessment as an official record for that program. CCAPs can be revised at any time and re-submitted to the Office of Academic Affairs.
7. *Collect and Store Data.* Data, including evidence of student learning, must be stored for reference.
8. *Plan Assessment Review.* Departments may design their own method for reviewing assessment data including the use of Assessments Days, where faculty come together to organize student work artifacts and discuss findings. Each review may focus on only a portion of the CCAP (e.g. two outcomes rather than all).
9. *Make Improvements Based on CCAP Reviews.* Data from student learning assessments are aggregated and results are shared among program faculty for discussion. These results can be used to identify patterns in student performance, including strengths and weaknesses. Faculty can discuss possible explanations for these findings and make changes as appropriate.
10. *Prepare and Submit Annual Reports.* Each year, departments are responsible for preparing and submitting an annual report that details any action(s) that was taken based on CCAP findings. These reports are submitted to the Office of Academic Assessment and a central repository for all Annual Reports will be maintained there.

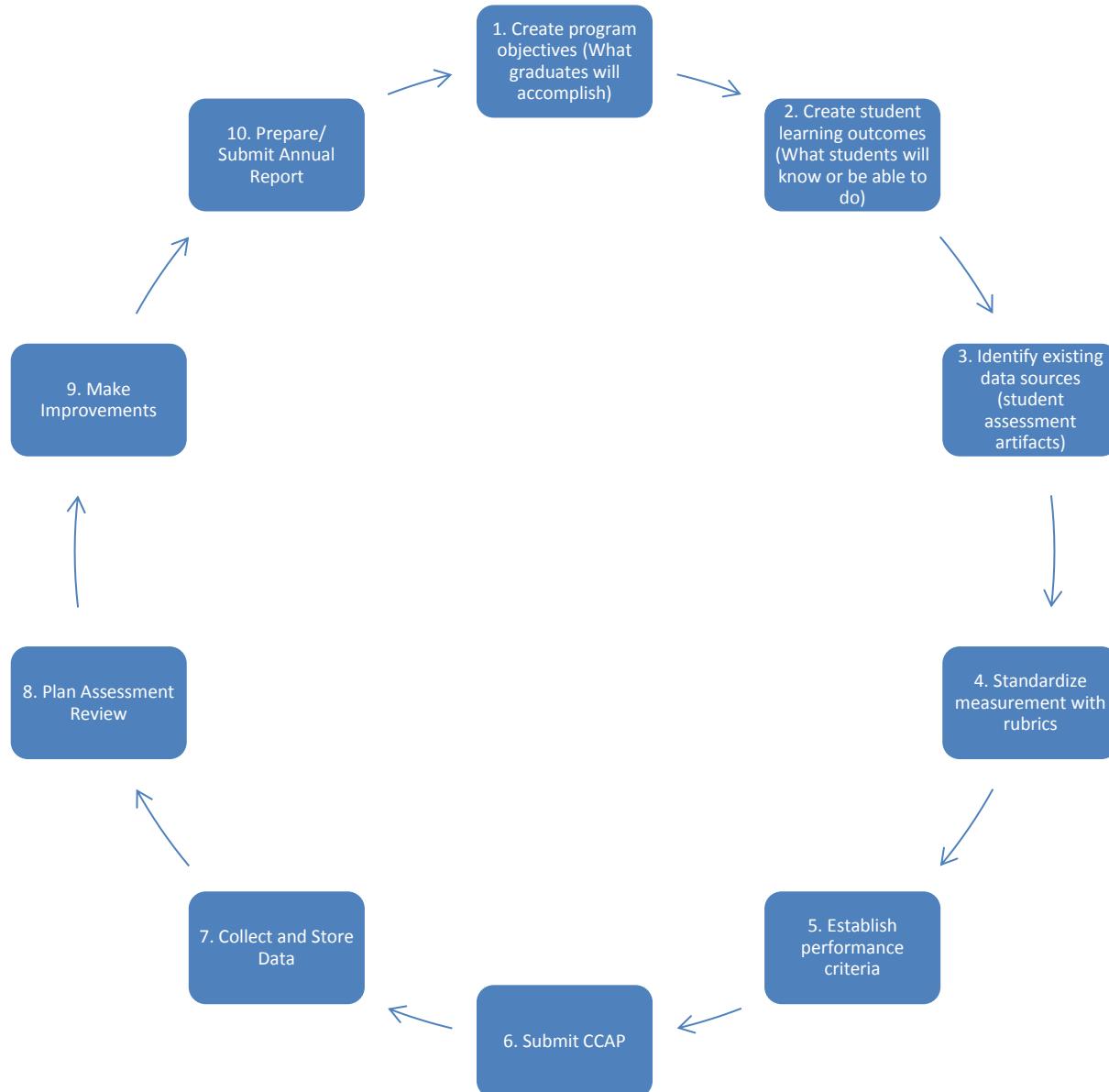


Figure 1: Process for CCAP Creation and Utilization