What is a Student Learning outcome (SLO)?
A SLO is the description of a task which successful completion is guaranteed to be evaluated at least once. For consistency’s sake, it follows these guidelines:

1. A single sentence.
2. Begins with an action verb describing the task. See Bloom’s taxonomy for helpful suggestions.
3. It covers about 3 hours of learning activities (class, lab, assignment).

Bloom’s Taxonomy of Learning
Here is a non-inclusive list of suitable verbs to articulate a SLO:

Knowledge: define, describe, identify, label, list, match, memorize, point to, recall, select, state

Comprehension: alter, account for, annotate, calculate, change, convert, group, explain, generalize, give examples, infer, interpret, paraphrase, predict, review, summarize, translate

Application: apply, adopt, collect, construct, demonstrate, discover, illustrate, interview, make use of, manipulate, relate, show, solve, use

Analysis: analyze, compare, contrast, diagram, differentiate, dissect, distinguish, identify, illustrate, infer, outline, point out, select, separate, sort, subdivide

Synthesis: blend, build, change, combine, compile, compose, conceive, create, design, formulate, generate, hypothesize, plan, predict, produce, reorder, revise, tell, write

Evaluation: accept, appraise, assess, arbitrate, award, choose, conclude, criticize, defend, evaluate, grade, judge, prioritize, recommend, referee, reject, select, support

Procedure
1. Gather a topic list for a course. This is likely to be found in the syllabus.
2. For each topic, jot down on which basis a student will be evaluated in general. Use the Bloom’s taxonomy above to articulate the statements. For example: “Enumerate the 5 phases of the project management cycle.”
3. You should get 10-20 of such statements for a 1 term course. These statements are SLOs.
4. To assist the curriculum map editor, provide 1 or a few keywords (tags) for each statements. This is helpful later for browsing and searching.
5. If applicable and for each statement, list 1+ SLO(s) that must be mastered prior to undertaking this SLO.

Overall time to map a course: 30 to 60 minutes.

What will be done with your SLOs
They will be entered in a Daedalus site where SLO and courses will be associated. The links provided in step 5 of the procedure will permit students and instructors to understand the pedagogical connections between courses. The keywords provided in step 4 will be used to allow people to explore a curriculum by theme instead of by course numbers. Ultimately, gathering and maintaining this data will make the process of program review simple, and will allow students, faculty and staff to get a handle on the curriculum by simply browsing through a website.
SCI 1101 - Computer Science II

Prerequisite Courses
- CSCI 1100

Assumed Learning Outcomes
What the student enrolling in this course is assumed to be able to do.
- Design (reusable) functions to divide the solution of a problem into simpler steps. [CSCI 1100]
- Explain and use the concept of arrays to store aggregate data. [CSCI 1100]
- Explain memory, binary numbers, and memory addresses. [CSCI 1100]
- Explain and use the concept of procedures to develop modularized code. [CSCI 1100]
- Describe the procedural programming paradigm and describe commonalities among a variety of commonly used procedural languages. [CSCI 1100]
- Write a reasonably-complex (200-250 line) modular procedural program. [CSCI 1100]

Student Learning Outcomes
Student learning outcomes that are covered by this course.
- Define what is an abstract data type (ADT).
- Describe the fundamental components of an object-oriented program such as variables, control structures, functions (methods), compound types (classes), and references.
- Use basic data structures (linked lists, queues, stacks, trees).
- Differentiate between arrays, linked-lists, and other aggregate data structures.
- Explain and use the concept of recursion.
- Explain the difference between (and use of) class and instance fields.
- Implement a linked list data structure.
- Describe the binary tree data structure and binary tree traversals.
- Implement a program using both class and instance fields.
- Implement searching, sorting, and other operations on a linked list.
- Implement the stack, queue, deque, and list abstract data types (ADT) using arrays and linked-lists.
- Define classes that use accessors (getters) and mutators (setters), and that overload standard methods.
- Identify applications of the stack, queue, and list abstract data types (ADTs).
- Implement stacks and queues as arrays, arraylists and linked lists.
- Describe the object-oriented programming paradigm and describe commonalities among a variety of commonly used object-oriented languages.
- Develop complete applications comprising multiple classes that integrate various object-oriented programming concepts.
- Explain and use the concept of inheritance.
- Explain and use the concept of method overloading.