DeAnza College

SLO Assessment Cycle for CIS 15C

Data Structures SLO Modified: [04/21/2010]

Delia Garbacea's Team Members:

1. Cynthia Lee-Klawender (x8609) CIS
2. Clare Nguyen (x8461) CIS

Additional Team members not on list/notes about team:

Clare Nguyen, Cynthia Lee-Klawender,

Additional Notes:

Outcomes:

Outcome 1: Statement Modified: []

Read, analyze and explain advanced C programs.

Assessment Cycle Records:

Outcome 1: Assessment Planning Modified: [06/04/2010]

Assessment Strategy Used:
Quarter: Winter 2010
Assessors: Delia Garbacea
Assessment Tools: Exams
Sections being assessed: 01, 01H

Outcome 1: Reflect & Enhance Modified: [06/09/2010]

Number of people involved in Phase III: 1

Changes:

Methods:
The methodology for assessing the outcome was a final exam question.
Given a program, students had to trace the steps of a recursive function and show the output.

Summary:
The class average for this question is 8.3
58% of the students obtained very good to excellent results, 27 % - good, 11% - satisfactory, and 4% failed.

1 - 0
2 - 0
3 - 0
4 - 1
5 - 1
6 - 2
7 - 3
8 - 4
9 - 8
10 - 7 Average: 8.3

This demonstrates that the students developed the skills they need to read, understand and explain advanced programs, which are necessary for developing and testing their own code.

Enhancement (Part I):
Although most of the students obtained good to excellent results, some of them need more practice.

Enhancement (Part II):

Outcome 2: Statement Modified: [09/08/2010]

Design solutions for advanced problems using appropriate design methodology incorporating advanced programming

Outcome 2: Assessment Planning Modified: [06/03/2010]

Assessment Strategy Used:
Quarter: Winter 2010

constructs.

Assessors: Delia Garbacea
Assessment Tools: Exams
Sections being assessed: 01, 01H

Outcome 2: Reflect & Enhance Modified: [06/10/2010]

Number of people involved in Phase III: 1

Changes:

Methods:
The methodology for assessing the outcome was a final exam question.
Given a project, students had to design a solution and explain their design.

Summary:
The class average for this question is 7.0.

27% of the students obtained very good to excellent results, 43 % - good, 15% - satisfactory, and 15% failed.

1 - 1
2 - 1
3 - 1
4 - 1
5 - 1
6 - 3
7 - 5
8 - 6
9 - 4
10 - 3 Average: 7.0

The majority of the students were able to design solutions for advanced problems. For most of the programming assignments there were in class exercises to create and analyze different designs. The students had to choose one of these designs and finish it using the appropriate design tools, or come up with another one. However, there is a tendency to write the code first, and then finish the design.

Enhancement (Part I):
To improve learning and encourage students to spend more time designing, the submission of the programming assignment will consists of two parts: first, the students will have to turn in the design (a draft), and then they will have to turn in the completed assignment and the updated design. Also, more design exercises (no coding) will be assigned throughout the quarter, and one such exercise will be given on the midterm exam too.

Enhancement (Part II):

Outcome 3: Statement Modified: []

Create and analyze efficiency of advanced level algorithms, code, document, debug, and test advanced level C/C++ programs using multiple source and header files.

Outcome 3: Assessment Planning Modified: [06/09/2010]

Assessment Strategy Used:
Quarter: Winter 2010
Assessors: Delia Garbacea
Assessment Tools: Exams • Programming Assignment
Sections being assessed: 01, 01H

Outcome 3: Reflect & Enhance Modified: [06/09/2010]

Number of people involved in Phase III: 1

Changes:

Methods:
The methodology for assessing the outcome was a final exam question and a programming assignment.

The final exam question was to create an algorithm for a given problem, then write a documented C function. Debugging and testing was not assessed since computers were not used during the final examination.

The programming assignment #3 out of 5 was used for assessing all parts of this SLO. The design was given in class, and the students had

two weeks to create algorithms, implement, debug and test them. Final program consisted of two or three source files and a header file.

Summary:
The class average for the final exam question is 6.6. 35% of the students got very good to excellent results, 31 % - good, 11% - satisfactory, and 23% failed.

1 - 3
2 - 3
3 - 0
4 - 0
5 - 0
6 - 3
7 - 4
8 - 4
9 - 5
10 - 4 Average: 6.6

Findings and conclusions for the programming assignment #3 out of 5.
The class average for the programming assignment is 16.0

27% of the students obtained excellent results, 47 % - good to very good, 13% - satisfactory, 3% failed, and 10% did not turn in the assignment. Out of 30 students, 3 students did not submit the assignment, 3 had a 2 points penalty for late submission; 7 students asked for more time to complete the assignment, but two were not able to finish the assignment. One student was asked to resubmit the assignment, but he didn't do it.

8 - 20points
6 - 19
5 - 18
3 - 17
2 - 16
2 - 14
1 - 5
3 - 0 - did not submit the assignment Average: 16.0

74% of the students were able to create algorithms, code, document, debug, and test advanced C programs.

Usually students are eager to write code, debug and test it, but reluctant to write documentation. Assignments without proper documentation were return to the student without a grade: they were asked to write proper documentation and resubmit the assignment. For this programming assignment, most of the programs were properly documented; no student was asked to rewrite the documentation; however, some points were deducted for inaccurate comments.

As expected, when given more time and a computer for debugging and testing, students provide better code than when writing code on paper, under stricter time constrains: 16.0 (maximum score 20) compared to 13.2, the equivalent for 6.6 (maximum score 10).

Enhancement (Part I):
Students demonstrated that they are able to create algorithms and write advanced C code, but they have to work more on enhancing their debugging and testing skills. They spend too much time debugging their programs, and not sufficient time testing them.

To improve learning and encourage students to spend more time testing they will be asked to create test plans and submit them with their assignments. To spend less time debugging, the students will be encouraged to use the incremental development of a program and learn different debugging techniques, including using a debugger.

Enhancement (Part II):

[Number of Outcomes for CIS 15C: **3**]