Course Assessment for Dummies!

(AKA: Measuring Student Learning in 6 Easy Steps)

Step #1: *Find your course syllabus and locate the learning outcomes listed for your course (eg. These are the ideas/concepts/skills that you want your student to have learned upon completion of your course). Check off the outcomes that you will be measuring on your Final Exam, Final Project, or Final Portfolio.*

Step #2: *In the table provided, list each Learning Outcome that you will test on the Final Exam (or Final Project or Portfolio). Record the problem number from the Final Exam that tests or measures how well a student mastered this objective.*

Step #3: *After the exam, record the total # of students taking the exam. Then, for each learning objective, tally up the number of students who demonstrated FULL, PARTIAL, or NO mastery of the material. From these tallies, calculate the percentage of students who demonstrated FULL, PARTIAL, or NO mastery of the material.*

Step #4: *Write up your SUMMARY OBSERVATIONS about what your data has shown you regarding how well the students learned the material. Where did they excel? Where were they weak? Then write up your ACTION PLAN for how you will change your teaching, course content, and/or evaluation measurement the next time you teach the course to improve student learning.*

Step #5: *Submit your results to your Curriculum Coordinator in* ***electronic form*** *and within one week of submitting your grades. (If you put off doing this assessment, there's a strong likelihood that it will never get done. I know this from first-hand experience!)*

Step #6: *Accept our thanks for your efforts and know that your students will benefit from the things you learned from doing this assessment!*

Sample of Step #1: ***(Find your course syllabus and extract the learning outcomes. Check off the outcomes that you will be measuring on your Final Exam, Final Project, or Final Portfolio.)***

CS152L: Computer Programming Fundamentals in JAVA For CS-Majors
Spring 2007

LEARNING OUTCOMES

A student in this course should be able to discuss the following topics, recognize proper and improper use of these items in existing code, and create new programs using the following items effectively.

- Introduction
  - Problem solving, algorithms
  - Computer organization, programs, compilers

- Basic Java
  - Intro to Abstract Data types
  - Primitive data types
  - Simple input and output
  - Program structure

- Decision-Making and Repetition in Problem Solving
  - General Selection and Conditions: if/else X
  - Iteration and looping: while, do/while, for X
  - Nested Loops X

- Object Oriented Programming Techniques
  - Classes X
  - UML diagrams X
  - Class methods
  - Method overloading X
  - Inheritance X
  - Polymorphism/Abstract Classes X
  - Interfaces X

- Data Structures in Java
  - 1-D arrays X
  - Multi-D arrays X
  - Arrays of objects X

- Program Development
  - Software development cycle
  - Object decomposition
  - Top-down design and functional decomposition
  - Information sharing between modules using parameters
  - Code testing techniques; good selection of test data
  - Code debugging techniques; tracing
  - Efficiency of algorithms
  - Object-oriented design and coding approaches

X – represents a Learning Outcome that will be tested on the Final Exam. Other outcomes listed were tested on either the Midterm Exam or through their bi-weekly Programming Assignments.

Sample of Step #2: ***(In the table provided, list each Learning Outcome that you will test on the Final Exam (or Final Project or Portfolio). Record the problem number from the Final Exam that tests this objective.***)

OUTCOMES ASSESSMENT
CS152L: PROGRAMMING FUNDAMENTALS FOR CS-MAJORS (JAVA)

Date: Spring Semester, 2007
Total # of students taking final exam:

FINAL EXAM ANALYSIS:
Most of this exam focuses on concepts taught during the second half of the semester. However, because each topic in the class builds on previous content, some of the earlier concepts are implicitly tested as part of evaluation of newer concepts.

Learning Outcome	Problem # on Exam	# Students demonstrating Full Mastery	# Students Partial Mastery	# Students No Mastery	% Students Full Mastery	% Students Partial Mastery	% Students No Mastery
Looping: For-Loops	3.c						
Nested Loops	4a						
Selection: If-else	5a						
Arrays (1D)	3						
Arrays (2D)	4a						
Arrays of Objects	5						
Interfaces	4b						
Inheritance	6						
Abstract Classes	8						
Searching	7						

(Note: if you want to include more information under the Learning Objectives column, you may find it easier to prepare this table in Landscape Mode.)

Sample of Step #3: ***(After the exam, record the total # of students taking the exam. Then, for each learning objective, tally up the number of students who demonstrated FULL, PARTIAL, or NO mastery of the material. From these tallies, calculate the percentage of students who demonstrated FULL, PARTIAL, or NO mastery of the material.)***

OUTCOMES ASSESSMENT
CS152L: PROGRAMMING FUNDAMENTALS FOR CS-MAJORS(JAVA)

Date: Spring Semester, 2007
Total # of students taking final exam: 12

FINAL EXAM ANALYSIS:
Most of this exam focuses on concepts taught during the second half of the semester. However, because each topic in the class builds on previous content, some of the earlier concepts are implicitly tested as part of evaluation of newer concepts.

Learning Outcome	Problem # on Exam	# Students demonstrating Full Mastery	# Students Partial Mastery	# Students No Mastery	% Students Full Mastery	% Students Partial Mastery	% Students No Mastery
Looping: For-Loops	3.c	12	0	0	100%	0	0
Nested Loops	4a	11	1	0	92%	8%	0
Selection: If-else	5a	12	0	0	100%	0	0
Arrays (1D)	3	11	1	0	92%	8%	0
Arrays (2D)	4a	11	1	0	92%	8%	0
Arrays of Objects	5	9	3	0	75%	25%	0
Interfaces	4b	9	3	0	75%	25%	0
Inheritance	6	4	8	0	33%	66%	0
Abstract Classes	8	3	9	0	25%	75%	0
Searching	7	6	6	0	50%	50%	0

Sample of Step #4: ***(Write up your SUMMARY OBSERVATIONS about what your data has shown you regarding the students' learning outcomes. Where did they excel? Where were they weak? Then write up your ACTION PLAN for how you will change your teaching/course/evaluation the next time you teach the course to improve student learning.)***

SUMMARY/OBSERVATIONS:

1. Students did better on the material we covered during the first half of the semester because it is conceptually easier, and because they continue to use it throughout the semester. It's very familiar by the end of the semester. These topics would be the first 3 learning objectives listed. The material covered at the end of semester is newer and students haven't programmed with it as much. I would predict that they would not be as proficient with this material, and the exam results support this prediction.

2. Even though some of the students are tallied as having only partial mastery of the newer material, many of them made only minor errors (syntax, etc.) but knew the concepts fairly well.

3. The topic where most students made some error was with Abstract Classes, although we did spend a good deal of time on this subject. It comes late enough in the semester that they don't get to program with it very much. Also, the question on the exam was a bit confusing because I was trying a new format of question. I 'm not sure my question assessed student learning as well as it might have.

4. Last semester my OA results indicated that I needed to focus more on the topic of Inheritance. I worked at incorporating the topic earlier in the semester and giving students more exercises and programs with the topic. The OA results above showed an improvement in student understanding on this topic – hopefully as a result of my response to the OA results from last semester!

5. I need to focus on searching more. Also, I'm not sure I made it clear that this topic would be on the final exam because I left it off the review list! Only some of the students studied this topic.

ACTION PLAN FOR NEXT SEMESTER:

1. I will work at introducing the topic of Abstract Classes earlier in the semester and giving students more in-class exercises to assess their understanding. I will also try to give them more opportunities to practice with this concept in the programming assignments.

2. I will continue to provide the extensive focus on Inheritance because it clearly paid off last semester.

3. I will continue to learn about creating a final exam that tests OA well. I need to work on separating some of the questions so that each focuses on only one topic; at this time, so many of the questions test multiple subjects, despite my improvements on the exams from previous semesters. It is an inherent problem with CS that so many of the topics are interconnected. However, with more topic separation on the exam questions, it would be easier to measure student proficiency in a particular area.

FAQ Page

Q1: What if I don't give a final exam in my course?
A: The University encourages that all instructors offer some means for evaluating overall student learning at the end of the semester. This might be in the form of a Final Exam, a Final Report, a Final Project, or a Portfolio. You may adapt the process shown above to create a rubric for measuring the appropriate Learning Outcomes in whatever measuring tool that you use.

Q2: I have quite a few Learning Outcomes. Must I measure all of them on the Final Exam?
A: No. Your Final Exam (or Report or Portfolio…) might measure only some of your original Learning Outcomes. You may have used other measurement tools throughout the semester to measure the remaining Learning Outcomes. We suggest that you follow the process above for each of your exams so that your accumulated data will demonstrate the overall course assessment. In some cases, some outcomes are measured by means other than exams such as essays, projects, speeches, labs, programs, reports, etc.

Q3: My Final Exam only has 5 questions. How can I measure all my Learning Outcomes in only 5 questions?
A: See if you can massage your Final Exam to include more questions, with each question focusing on a narrower topic than before. In my earlier exams, I only had 5-6 questions on the exam and each question either had many parts or covered several topics. With a bit of practice (and much tweaking!), my exam evolved to include 10-12 questions that were each more focused on a limited topic. This made the assessment easier.

Q4: When do I need to collect this data?
A: Ideally, it should be done at the end of every semester while you are in the process of grading your Final Exam. Please submit your Assessment information to your CC electronically within a week of submitting your grades. This will assist your CC in meeting his/her deadlines for creating Outcomes Assessment reports for the curriculum area.

Q5: How do I distinguish between FULL, PARTIAL, and NO mastery of the material?
A: There is no magic formula here and you need to use your best guesstimate. For example, if an exam question is worth 6 points, I usually count FULL mastery for a student who earns 5 or 6 points on the question. PARTIAL mastery would be tallied if a student earns 3-4 points. If a student earns fewer than 3 points, then he/she didn't really master the material at all. This is where it becomes easier to tally if the exam questions are fairly well focused on just one topic. This is a tough issue, but I think (hope!) I get better at this each semester.

Q6: How do I evaluate comprehensive assignments, such as projects and term papers?
A: For assignments or questions that may encompass several of your learning objectives, you may find that it helps to develop rubrics for grading. A rubric may encompass several sub-objectives; it defines what it means for each one to show full, partial or no mastery. For example, a simple rubric for, "Accuracy of content" might look like this:

Full Mastery: Information (names, facts, description of events or data or theories, etc.) included in the presentation is consistently accurate.
Partial Master: No significant errors made. Errors only in difficult, obscure areas.
No Mastery: Significant errors in basic information, even if some information is accurate.