End of Course Assessment Memo
CS216 – Program and Data Representation
Ruth Anderson (fall 2003)

Course Objectives:

1. Implement basic and advanced data structures in C++ (stacks, queues, hash tables, binary search trees & balanced trees, heaps, graphs).

2. Evaluate the asymptotic time and space complexity of the above listed data structures and assess the suitability of a data structure for a particular problem.

3. Develop an understanding of the underlying representation of basic data structures (ints, chars, floats, arrays, records, pointers) as well as more advanced structures (linked lists, 2-D arrays, trees, hash tables, heaps, graphs).

4. Implement basic program control and data structures in an assembly language (loops, conditionals, subroutines and parameter passing modes, arrays)

5. Develop a basic understanding of program execution model and underlying computer hardware and software (fetch-execute cycle, memory hierarchy, operating system, compiler).

Mapping of Course Objectives to ABET Outcomes:

Course Objectives	a) an ability to apply knowledge of mathematics, science, and engineering	(b) an ability to design and conduct experiments, as well as to analyze and interpret data	(c) an ability to design a system, component, or process to meet desired needs	(d) an ability to function on multi-disciplinary teams	(e) an ability to identify, formulate, and solve engineering problems	(f) an understanding of professional and ethical responsibility	(g) an ability to communicate effectively	(h) the broad education necessary to understand the impact of engineering solutions in a global and societal context	(i) a recognition of the need for, and an ability to engage in life-long learning	(j) a knowledge of contemporary issues	(k) an ability to use the techniques, skills, and modern engineering tools necessary for engineering practice
1	X	X			X						X
2	X		X								X
3							X				
4	X		X		X						
5											X

Assessment Tools:

Student achievement of the course objectives is assessed by 9-10 programming assignments (pre-lab, in-lab, and post-lab activities) as well as with two two-hour in-lab exams and a 3-hour comprehensive final exam. I also gave an on-line survey using SurveySuite mid-semester seeking to identify which components of the course were most useful in learning (lab, lecture, textbook, which labs).

A description of the assessments tools that were analyzed for this report is given below:

Objective 1: Lab #1 – implement a doubly-linked list, Lab #2 - implement and use stacks, Lab #4 – modifications to binary search trees and AVL tree implementations, use of binary search trees, Lab #7 – implement and use hash tables, Lab #8 – use heaps to implement Huffman coding. Questions about these data structures from the final exam will also be used.

Objective 2: Done on several exams and labs. This assessment will use questions from the final exam.

Objective 3: Lab #3 – investigates these issues with an in-lab activity using the debugger. This assessment will use questions from the final exam.

Objective 4: Lab #5 – students implement several programs in a simple assembly language, IBCM, Lab #6 – students implement several programs in x86 assembly language and do a detailed analysis of how various aspects of C++ program control and data are implemented in x86 by the compiler. We will also use questions from the final exam for this assessment.

Objective 5: This is handled in part in several labs. For this assessment we will use questions from the final exam.

Percentage of Students Meeting Course Objectives:

		% of students meeting objective				
Coursework	Topic	Objective 1	Objective 2	Objective 3	Objective 4	Objective 5
Lab 1	Linked lists	83				
Lab 2	Stacks	91				
Lab 3	Data Representation			94		
Lab 4	BSTs & AVL trees	89				
Lab 5	IBCM				97	
Lab 6	X86				87	
Lab 7	Hashing	94				
Lab 8	Huffman Coding	89				
Final Exam	Comprehensive	69	49	89	89	31
Average		**86**	**49**	**91**	**91**	**31**

Note: "Meeting Objective" was defined as obtaining over 75% of the possible points.

General Evaluation in terms of Course Objectives:

Objective 1: Overall students seem to be meeting this objective fairly well. Several labs give them practice in implementation.

Objective 2: Students do o.k. with this but could be doing better. Although it should be noted that using a few questions on the final exam as the only assessment of this objective is probably not the most accurate measurement. In the future, results of questions on earlier exams and labs should be used. Since this is the first time students have seen this material, it is not surprising that it is one of the weaker areas. Computer Science majors will see it again in cs432. Nevertheless, the scores on the final exam for these questions could be higher. I will try to incorporate more practice questions in this area in the future.

Objective 3: Students are doing fine with this objective. Lab 3 and some of Lab 6 addresses this issue.

Objective 4: Students are doing fine with this objective. The labs give a bit of practice. In addition I spend a reasonably large amount of class time going over examples which students indicate on my mid-semester feedback surveys is very helpful in learning the material.

Objective 5: Students are doing o.k. with this objective, but could be doing better. It should be noted that only a very small number of questions from the final exam were used as the assessment measure for this objective. In the future, results of questions on earlier exams and labs should be used. This is not a major objective of the class, or rather the level of understanding is not expected to be incredibly deep, and so this percentage is not of major concern. 74% of the students got over 50% of the points on these questions from the final exam. Nevertheless this could be improved in the future.