DO FUTURE PROFESSORS NEED TEACHER TRAINING?

Charles Bauer[1], Matthew Bauer[2]

Abstract *– Most university professors consider their first teaching experience a disaster. After lecturing for several weeks, the first examination results quite often are shocking in revealing the lack of knowledge of the students. At Illinois Institute of Technology we offer two graduate courses that can make this first venture into the classroom a more rewarding experience for both the professor and the student. Planning, developing and teaching a course requires three equally important components; 1) knowledge of the subject matter, 2) course content organization, and 3) course material presentation. These three components have been determined by questioning many undergraduates and graduate students and asking them to list single words which best describe the outstanding professors they have encountered. The second and third of these items are addressed by our two courses*

Background

For many years the Computer Science Department of Illinois Institute of Technology (IIT) has offered a Master of Science for Teachers (MST) in the teaching of Computer Science. Two of the courses (CS 560 and CS 561) in the program have been found to be of great help to Ph.D. students nearing the end of their student days.

The three equally-important components of a successfully taught course at any level are a) knowledge of the subject matter, b) logical organization of the concepts to be presented, and c) a presentation style that keeps both the instructor and the student interested, and gives the student the best opportunity to learn. If all goes well, an eagerness to learn on the part of the student is developed.

In the case of the Ph.D. student, knowledge of the subject matter is assumed.

CS 560

The technique used to organize the subject material is straightforward and uses the computer to help plan the organization of the concepts to be covered.

Step one is to determine how much class time is available. A typical three credit hour semester course (3-0-3) (all lecture, no lab) will meet for 15 weeks with a 16th week for the final examination. This provides the instructor a total of 45 academic hours of instruction. (Academic hours are 50 minutes.) A (2-2-3) course, two hours of lecture and lab per week, has 60 hours on instruction time for the semester. This limit on hours on instruction is the one part of the course that is fixed. Everything else is determined to some degree by the instructor.

The next step is to list the major concepts that are to be covered in the course. Usually this means about five to ten concepts; each is then allotted a certain number of hours of instruction. These hours plus any examination time, other than the final examination, must add up to the total for the course. A spreadsheet of weeks and hours is then developed.

Each concept is then sub-divided into smaller concepts, and the final step is to complete weekly information, consisting of a cover sheet with pages to be covered from the text, objectives for the week, and concepts with sub-concepts to be covered. This is followed by all of the lecture notes, hand-outs, and any other material for the week.

The compilation of this material usually takes about eight weeks to develop and refine. Much reorganization takes place as the material for each week is developed. Time limits for concepts are changed, and reorganization of the concepts quite often occurs.

CS 561

Presentation techniques have changed since the days of classroom blackboards and chalk. But before presentation techniques are discussed some time is spent in a discussion of learning theories. The most important idea that must be understood by future instructors is that everyone in their class might learn in a different way. Providing for these differences is the most difficult task for any instructor.

Undergraduates are interviewed to determine which concepts in their Computer Science courses were the most difficult to understand. Class discussions of how this concept might be presented are debated, with several approaches coming from the discussion. CS 561 students are then asked to develop Single Concept Learning Modules on the web to provide undergraduates with a means to better grasp the concept. The development of the SCLM for a particular concept requires that detailed step by step presentation takes place.

Many classrooms today have facilities for the instructor to bring in a laptop which allows computer-based technique. Great presentation skills take time to develop.

[1] Charles Bauer, Illinois Institute of Technology, Computer Science Department, Chicago, Illinois 60616, bauerc@iit.edu

[2] Matthew Bauer, Illinois Institute of Technology, Computer Science Department, Chicago, Illinois 60616, bauerm@iit.edu

Each instructor must develop their own classroom methods and continually try to improve their presentation techniques. One excellent method to improve is to video tape a lecture and the review the tape critically.

Students who have completed both courses and are now started on their careers have sent back reports that their initial experiences have been favorable.

The next pages have samples of the material developed in CS 560, and a partial list of SCLM developed in CS 561. More information can be found on:

www.cs.iit.edu/~cs560 www.cs.iit.edu/~cs561

Summary

CS 430 (3-0-3)
Introduction to Algorithms

Text

Cormen, Leiserson, Rivest, *Introduction to Algorithms*, MIT Press/McGraw-Hill, 1990

Objectives

- Understand and apply algorithmic analysis (both space and time) for both non-recursive and recursive algorithms
- Understand and apply the advanced data structures of heap, hash, graph and balanced binary search trees
- Understand and apply the theory of the algorithmic approaches of divide-and-conquer, greedy, dynamic programming

Concepts

1.	Algorithm Basics	3 hours
2.	O-notation, Growth of Functions, Summations	3 hours
3.	Non-Recursive Sorting	3 hours
4.	Recurrence Relations	3 hours
5.	Recursive Sorting	3 hours
6.	Trees	1 hour
7.	Heaps, Heapsort	3 hours
8.	Exam #1	3 hours
9.	Hash Tables	3 hours
10.	Greedy Algorithms	4 hours
11.	Graphs	6 hours
12.	Dynamic Programming	4 hours
13.	Balanced Binary Search Trees	4 hours
14.	NP Basics	1 hour
15.	Parallel Algorithm Basics	1 hour
		45 hours

SINGLE CONCEPT LEARNING MODULES

COURSE	TOPIC
GENERAL:	Searching Sorting
CS105 INTRODUCTION TO COMPUTER PROGRAMMING I	Problem solving with Pseudocode1 Problem solving with Pseudocode2 Boolean expressions and truth tables Iteration Arrays1 Functions Functions2 File Streams (HTML from C++) Pointer
CS106 INTRODUCTION TO COMPUTER PROGRAMMING II	Recursion Template Functions
CS200 INTRODUCTION TO C++ PROGRAMMING	Problem solving with Pseudocode1 Problem solving with Pseudocode2 Boolean expressions and truth tables Iteration Arrays1 Functions Functions2 Recursion
CS331 DATA STRUCTURES AND ALGORITHMS	Recursion Queues Singly Linked List Sets
CS350 COMPUTER ORGANIZATION ASSEMBLY LANGUAGE PROGRAMMING	Boolean expressions and truth tables Addressing Modes Addressing Modes2 Fetch and Execute Cycle Pop vs Push Branch and Jump Encoding
CS425 DATABASE ORGANIZATION	Relational Algebra
CS441 JAVA	Concurrency Programming
CS450 OPERATING SYSTEMS	Deadlock Disk Scheduling Algorithms Fork System Call Piping and Redirection Semaphore Socket Programming Job Scheduling Memory Management
CS487 SOFTWARE ENGINEERING I	Dataflow Coupling Cohesion