Programming Competitions as a Basis for an Algorithms and Data Structures Course

John Paxton

Montana State University

CCSC - NW
Friday, October 12, 2007

Outline

I. Course Overview
II. Assessment
III. Curriculum Ideas
IV. Discussion
I. Course Overview

• http://www.informatik.uni-leipzig.de/~paxton/algorithmics

Background

• Practical Applications of Data Structures and Algorithms
• The University of Leipzig
• Winter Semester 2006 / 2007
• 75 upper division students
• 4 credit, elective course
Funding

• Fulbright Award
• Montana State sabbatical

Course Goals

• Improve problem solving abilities
• Improve (Java) programming skills
• Improve technical English abilities
Course Format

- 90 minute lecture
- 90 minute lab
- Weekly exercise (30%)
- Local programming contest (10%)
- Final exam (60%)

Local Programming Contest
Course Topics

- Simple Input / Output
- Base Conversions
- Large Numbers
- Combinatorics
- Strings
- Sorting

Course Topics

- Grids
- Trees
- Graphs
- Dynamic Programming
- Artificial Intelligence
II. Assessment

- Student
- Instructor

Student Assessment

1. The course took a practical approach to problem solving. How did you like this approach?

0 – strongly dislike
1 – dislike
0 – neutral
12 – like
8 – strongly like
2. The majority of the lecture time was spent coding solutions to problems. How valuable was this technique?

   2 – not valuable
   5 – somewhat valuable
   10 – valuable
   4 – very valuable

3. How much did your problem solving skills improve during the course?

   0 – none
   5 – a little
   9 – some
   7 – a lot
Student Assessment

4. How much did your programming abilities improve during the course?

0 – none
7 – a little
11 – some
3 – a lot

Instructor Assessment

• Emphasize testing more
• Post alternate solutions
• Use a grading script
• Have students take final individually (pairs underperformed by 12.5%)
• Have students code in class
• Improve the assessment process
III. Curriculum Ideas

- Problem solving course
- Programming contest course
- Study abroad course
- Integrate into existing courses (CS I, CS II, artificial intelligence, networks, etc.)

IV. Discussion

- How to avoid a “hack-fest”
- Managing 75 students with no TA
- A peek at the website
IV. Discussion

• Thank you!

• Any questions?