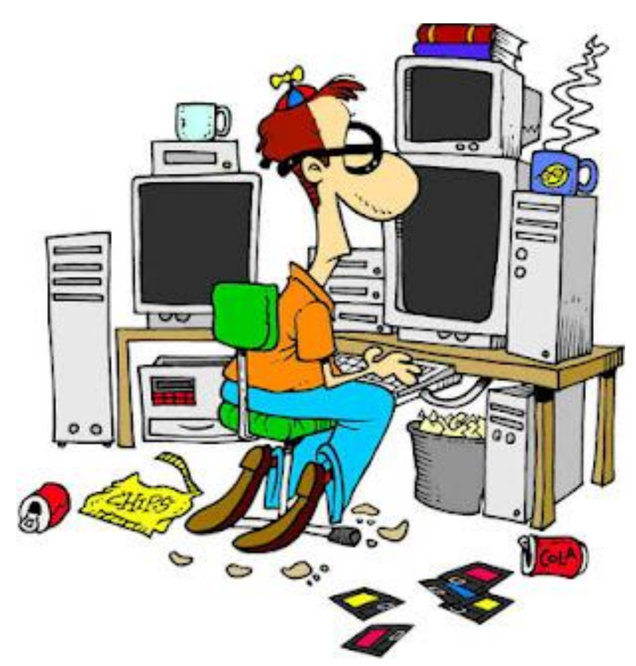
Adapting the CLASS for Use in Computer Science

Allison Elliott Tew[1], Brian Dorn[2], and Oliver Schneider [1]-- aetew@cs.ubc.ca, bdorn@hartford.edu, oschneid@cs.ubc.ca

[1]Department of Computer Science, University of British Columbia; [2]Departments of Computer Science and Multimedia Web Design and Development, University of Hartford

BACKGROUND AND MOTIVATION



The Computing Attitudes Survey (CAS) is a newly designed instrument, adapted from the Colorado Learning Attitudes about Science Survey (CLASS), for measuring novice to expert-like perceptions about computer science. The goal of the instrument is to address a wide variety of student beliefs about learning computer science and the nature of computer science knowledge. Following an iterative design process, validation and pilot testing began Fall 2011 and continued with pre/post term deployment Spring 2012.

INSTRUMENT DESIGN

Items in the CAS were developed using three strategies: direct replication from the CLASS, adaptation from the CLASS, and generation of CS specific statements.

REPLICATION

Many items (n = 29) in the CAS were able to be used as is, with the word “physics” changed to “computer science.”

Q1: After I study a topic in computer science and feel that I understand it, I have difficulty solving problems on the same topic.

Q3: I cannot learn computer science if the teacher does not explain things well.

ADAPTATION

A few statements (n = 4) in the CAS were able to be adapted using terms more appropriate to the discipline. For example, substituting algorithms for equations.

Q33. When I am working on a computer science program, I try to decide what reasonable output values would be.

Q41. Spending a lot of time understanding where algorithms come from is a waste of time.

CS SPECIFIC STATEMENTS

Other statements were not applicable and were replaced by discipline specific statements (n = 26). CS specific topics include abstraction, errors and testing, and problem solving strategies.

Q8. When working on a complex computer science problem, I have to understand all of the details of the program implementation before I am able to make progress on a solution.

Q25. I find the challenge of solving computer science problems motivating.

VALIDATION PROCESS

CAS items were modified, added, or removed based on faculty and student interviews, expert review, and factor analysis. Evidence of response validity for individuation statements was collected through:

- Faculty Think Aloud Interviews
- Student Think Aloud Interviews
- Expert Review for Consensus

THINK ALOUD INTERVIEWS

We conducted think-aloud interviews with faculty and students to verify that the wording and meaning of the items was clear and being interpreted consistently. During each interview, the participant first completed and submitted a version of the survey. They were then verbally presented each item and asked to respond and explain their response.

FACULTY INTERVIEWS: (n = 11)

Interviewed faculty from 8 different institutions, ranging from large, research university to small, liberal arts college

STUDENT INTERVIEWS : (n = 9)

Interviewed students, from 2 institutions, completing a first course in computing; included both a majors and non-majors approach to CS1

EXPERT CONSENSUS

The CAS has been given to 37 computer science experts from 28 universities around the world. We asked the faculty experts to rate each statement on a five point Likert-type scale, from Strongly Disagree to Strongly Agree.

Consistent expert responses were given for 45 of the 59 statements. If fewer than 66% of the experts did not agree or disagree to a particular statement, the statement was dropped from the instrument. The average consensus for all the statements on the final version of the CAS is 87.1%. These items will be used in scoring student responses to the CAS.

FACTOR ANALYSIS: PRELIMINARY RESULTS

We used an iterative, exploratory factor analysis to categorize the CAS statements into different aspects of student thinking. The goal of this analysis is to:

- Examine student thinking more deeply
- Derive the categories from actual student responses, rather than preconceived schemas from the researchers’ perspective

METHOD

- Data was transformed to 3-point scale (Agree with Expert, Neutral, Disagree with Expert).
- Remove students who already exhibit highly expert (>80% agreement) scores, as this will bias the resulting categories.
- Identify statistically valid categories using exploratory factor analysis. Replicating the CLASS analysis method, we used the principle components extraction method with a direct oblimin rotation.
- Analyze and revise the categories by evaluating both the statistical contribution of each statement as well the interpretability of the resulting factors.

CANDIDATE FACTORS

Exploratory factor analysis uncovered the following eight candidate factors:

Problem Solving: Helplessness	1, 4, 5, 6, 11, 14
Problem Solving: Strategies	1, 22, 40, 53
CS: Fixed Mindset	11, 41, 42, 47, 48, 52, 54, 56, 57, 58
Real World Connection	21, 34, 37, 51, 52, 55
Problem Solving: Confidence	24, 25, 51
Problem Solving: Reasoning	35, 38, 43
Abstraction	9, 35, 50
Personal Interest/Enjoyment	1, 6, 29, 30, 33, 34, 38, 40, 54, 59

NEXT STEPS

A full-scale pre/post term pilot study is current underway Spring term, 2012. Additional analysis will be needed to refine our preliminary findings, and additional data collection and analysis will be necessary to capture the diversity of content and pedagogical approaches currently addressed by introductory computing courses across the globe.

This work is supported in part by the CWSEI Carl Wieman Science Education Initiative at the University of British Columbia