Using a “Tech Interview” to Assess Communication Proficiency and Retention of Introductory Computer Science Concepts

Teagle Assessment Project

John P. Dougherty
Department of Computer Science
Haverford College
October 20, 2014

Overview

As is often the case in education, students are challenged to retain information learned early in the curriculum. We as teachers endeavor to ensure that this information is transferred from one course into all subsequent courses and, eventually, in practice after graduation. In computer science, it is often the case that students conflate software tools with concepts, or even programming with foundational material. Programming and tools are needed to deepen understanding of concepts in assignments and lab exercises (and address the initial challenge), but they are many times engaging enough in their own right to distract from the underlying foundation intended.

Furthermore, student interest in the computing industry is increasing, and many undergraduates are concerned about the interview that is expected for their first time out of college. Students want to succeed, and preparation for this interview (i.e., practice) is a motivation in its own right.

In order to assess the retention of concepts (as opposed to tools, programming) from the initial sequence of two courses, we chose to conduct concise “tech interviews” of all majors as part of their capstone thesis projects in computer science. We traded hard numbers (e.g., grades, scores, whether a program ran correctly, averages) for a more holistic assessment that involved a personal interaction where the entire thinking process of the student can be viewed and studied.

Our objective is to assess the impact of the computer science curriculum at Haverford College to prepare majors to understand, retain and communicate effectively about concepts introduced in the first two courses but utilized and emphasized in subsequent courses.

Methodology

---

We considered how industry conducts interviews, especially given the recent surge in student interest in computing careers, and the “legend” of the tech interview. This legend involves an intense, pointed set of questions from a team of technical people probing the interviewee about their understanding of programming. These students are tested as well as their abilities to think on their feet, to ask clarifying questions and otherwise interact with the tech team, and to communicate their answers effectively.

The first step was to identify the important concepts that we expect our student to retain and employ. The Department of Computer Science at Haverford has developed a set of learning goals that can be summarized into three broad categories:

1. Each student will realize their full ability to think deeply.
2. Each student will communicate their thinking clearly and effectively.
3. Each student will identify, interpret and evaluate the theoretical, practical and ethical implications of their work in the field.

The present assessment project is designed to gauge the effectiveness of the first two points. More specifically, we look to obtain evidence to assess point #2 (communication) and use this as an indirect way to gauge point #1 (conceptual understanding).

Next, a scoring rubric was developed to provide an explicit list of items to be assessed, as well as provide a means to evaluate each interview as consistently as possible. The rubric was partitioned into three parts:

- **content**: demonstrates knowledge of the material
- **connection**: demonstrates an awareness of the audience, adjusting accordingly
- **communication**: demonstrates professional presentation skills

The interviewer scoring rubric is found in the appendix of this document.

Our next step was to identify an interview question that could be used in the interview. This question had to balance a number of concerns in tension, including

- **appropriate**: would many of the concepts identified as in our goal be addressed?
- **accessible**: is this a problem that is not too difficult to understand?
- **concise**: can the problem description, work and solution(s) be accomplished in a timely manner?
- **robust**: are multiple approaches and solutions applicable, so the student is challenged to choose among a set of possibilities (and explain their choice)?

The question selected involved purchasing an item in various sized sets where the unit price per item varied, mandating that the students think dynamically and optimize under various

---

2 Dean Jackson. The Google technical interview: How to get your dream job. *XRDS 20, 2* (December 2013), 12-14.
circumstances using a simple arithmetic model and (typically) a single loop (or recursion). The script for this question is also available in the appendix of this document.

Each senior computer science major scheduled a meeting with the interview team. During the pilot phase in Spring 2013, that team consisted of Cathy Fennell, Cris Fuller and John Dougherty. Four students were involved in the pilot study.

However, for this actual study in the Fall of 2013, Pf. Dougherty conducted and recorded all interviews alone with the interviewee/participant using the Computer Science Lounge, KINSC H111. This change was mandated due to scale; we now have twelve majors in Computer Science (only ten were able to participate in the current project). It was a challenge to schedule people to meet (even though Pf. Dougherty was on sabbatical leave at the time). These issues of scheduling and scale are discussed at the conclusion of this document.

Each interview took approximately 12-20 minutes to complete, and was reviewed by Pf. Dougherty as well as a member of the Center for Career and Professional Advising (CCPA) staff at Haverford College. Each reviewer completed the scoring rubric, as well as provided comments. Pf. Dougherty, as a representative of the Department of Computer Science (CS) Department, completed the entire rubric for each student. CCPA staff members generally completed only the portion of the rubric involving connection and communication as is their expertise, though they were welcome to provide content feedback (and that did happen for three students). Furthermore, each student was reviewed once by CCPA and once by Pf. Dougherty. Other members of the Computer Science faculty are in the process of completing their assessment rubrics to provide a third review (i.e., a second technical view) of the interview.

By the end of the project, each interview could be completed in ten minutes. This time efficiency is important we scale this project to a growing number of computer science majors.

Results

Although the interview does not lend itself to the standard hard numbers found many times in (computer) science, we are able to present some general reports that get at a few of the assessment questions posed.
Quantitative

Figure 1 reports the mean score given to the ten student interviewed for the ten content assessment items as reported by CS/Pf. Dougherty. (Note: larger bars indicate a higher rating and a more positively scored demonstration; a link to the complete rubric item descriptions that expand the left vertical axis categories are found in the appendix.)

As only three students were assessed for content knowledge by the CCPA team, these results were set aside. The initial observation involves the items where scores were highest; they all seem to be focused on the more abstract items in computing, which is a stated intention of the program as these items are rarely replaced. This observation implies that more practical items are not as well demonstrated, and these are areas that are considered important to software developers and to industry (i.e., place to improve the CS curriculum).

We turn to communication in Figure 2, where both CS and CCPA people reviewed and scored each candidate. The first observation may be explained by experience bias; CS rates the the ten candidates higher in six of eight categories involving communication. Some of the largest

Figure 1: Content Knowledge Assessment by CS Faculty

Figure 2: Communication and Connection Assessments
gaps, “volume” and “enunciation,” may be explained possibly by the expected sound level for CS people, or it may be more of an issue with a recorded (CCPA) vs. an interactive/live (CS) interview. Still, the scores for expressiveness, pacing, and others were fine with the CS evaluation but not as expected by those outside of computing. This observation should inform both students and CS faculty as we prepare students to communicate with those outside the field, a stated learning goal.

It is also interesting that the students demonstrated more creativity to CCPA than witnessed by CS (or at least registered). Perhaps this observation can be explained again by experience, where presentation skills we use all the time in computing/software development are seen as novel by those outside of computing.

This assessment project was also looking for evidence to determine if concepts in the introductory two courses was retained and could be applied. Figure 3 plots each of the nine participants\(^4\) who completed both (or either\(^5\)) of the two introductory courses at Haverford or Bryn Mawr\(^6\) Colleges with the mean content score from the interview. The horizontal axis represents the mean of the (potentially) two grades in cs1 and cs2 (out of 4.0). Ideally, all student are represented at the upper right of Figure 3, indicating great grades and a fine interview. The expectation would be that points representing students fall on the diagonal line from origin to upper right, indicating a strong positive correlation. Roughly four points fall on this line, and one point above indicating a “better than expected” interview. The three points below this diagonal suggest that some content knowledge is not found as readily in the later interview as during the original courses. The student represented by the point at the lower left supports that a tough initial experience correlates to a similar interview performance.

\(^4\) one student was waived out of the first two courses after discussion with CS faculty \\
\(^5\) one student withdrew from cs1, then placed into cs2 after discussion with CS faculty \\
\(^6\) one student took the two introductory courses at Bryn Mawr College
Figure 4: Longitudinal Correlation for Communication and Connection

Figure 4 attempts to capture the same correlation between student grades in cs1/cs2 and their performance in the interview, this time for communication. As witness in Figure 2, we see that CS assessed the performance better overall. The general trend seems to remain as seen in Figure 3; namely, the sole outlier to the lower left remain for both CS and CCPA, and (too) many of the points lie below the expected diagonal.

It is difficult to see the distinction between CS and CCPA evaluations for individual students, motivating Figure 5 below.

<table>
<thead>
<tr>
<th>student</th>
<th>mean grade</th>
<th>CS</th>
<th>CCPA</th>
<th>Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>4.00</td>
<td>3.18</td>
<td>3.40</td>
<td>-0.22</td>
</tr>
<tr>
<td>1</td>
<td>4.00</td>
<td>2.73</td>
<td>3.13</td>
<td>-0.40</td>
</tr>
<tr>
<td>2</td>
<td>3.85</td>
<td>3.00</td>
<td>2.56</td>
<td>0.44</td>
</tr>
<tr>
<td>3</td>
<td>3.70</td>
<td>3.92</td>
<td>2.92</td>
<td>1.00</td>
</tr>
<tr>
<td>4</td>
<td>4.00</td>
<td>4.00</td>
<td>2.27</td>
<td>1.73</td>
</tr>
<tr>
<td>5</td>
<td>2.70</td>
<td>2.10</td>
<td>2.56</td>
<td>-0.46</td>
</tr>
<tr>
<td>6</td>
<td>4.00</td>
<td>2.55</td>
<td>3.36</td>
<td>-0.82</td>
</tr>
<tr>
<td>7</td>
<td>-</td>
<td>3.64</td>
<td>2.59</td>
<td>1.05</td>
</tr>
<tr>
<td>8</td>
<td>3.85</td>
<td>2.70</td>
<td>2.67</td>
<td>0.03</td>
</tr>
<tr>
<td>9</td>
<td>4.00</td>
<td>3.59</td>
<td>3.36</td>
<td>0.23</td>
</tr>
</tbody>
</table>

Figure 5: Individual Assessments for Communication and Connection

Figure 5 shows the difference between the scores report by CS and by CCPA people for the same interviewee, and includes their corresponding grades obtained in the initial course sequence. For example, student #6 earned 4.0 mean in the introductory courses, and received the second highest score from CCPA, but the second lowest score from CS in terms of
communication; at the same time, student #4 also averaged a 4.0 in the introductory courses, and received the highest score from CS but the lowest from CCPA.

These data observations are mutually extreme, and suggest that our students who seem to have the conceptual knowledge (as indicated by course grades) but connect unevenly during (pseudo-) professional presentations. That may suggest we need to provide a more diverse set of presentation experiences so that our students may receive more practice and more feedback to minimize this difference.

**Qualitative Assessments**

There are many observations to share here; let us start with those from the CCPA team of interviewers. These assessments can be roughly partitioned into the categories of “functionally effective/ineffective” and “distracting.” We are concerned with students where they apparently got in their own way, did not write down algorithms or ideas in a clear or readable manner, and/or did not practice enough “thinking out loud.” We are glad to have these observations as we can see that our students need more guidance about effective presentations in computing.

The less rigorous assessment provided by CS (*i.e.*, me) suggest that the students are listening and observing what they are getting in the courses, especially from *my* courses. The students generally start the session as one might expect in problem solving. The stronger students realize this is a chance to demonstrate what they know, and begin by asking a few questions to clarify (and also show they were listening). These better interviewees also gave a “birds-eye view” of the situation, and a few even noted more than one approach, including the trade-offs involved. CS did note that students had to be guided at times unexpectedly. For example, a few students had to be explicitly prompted about such basics as test cases, preconditions, and even efficiency/complexity. Once prompted, they would get on track and provide quality answers.

CS also noted a few of the students had difficulty seeing ways to generalize the problem. We would hope that such tools as test suites would help them see patterns sooner. Also, a few students chose to actually change the objective (within reason), often in a more useful way than stated in the problem. Vocabulary was not an issue, students did seem to understand and use terms appropriately.
Participant Survey Results

A concise but provocative survey was offered to participants almost a year after the project had concluded, and was completed by five (5) participants (50% survey participation). The following quick observations are made from this survey:

- about half the students felt they understood the goal of the interview
- four agreed that the question was representative of similar interview question that were used in other interviews; one reported “neutral”
- about the same response regarding the interview question as a vehicle to demonstrate their abilities and understanding of computing
- most report the time felt like it went fast during the interview
- when asked if they felt they succeeded in the exercise, two strongly agreed while two other simply agreed; one disagreed
- responses were mixed about appreciating multiple approaches to solving the problem during the interview
- most reported they felt they had enough opportunity to speak during the interview
- all participants reported feeling the interviewer understood their approach and proposed solution to the problem presented
- all either agreed (3), or strongly agreed (2), they believed that the Teagle project interview was good practice for their actual job interviews
- most students felt the space used for the interview was acceptable to use for this project

The raw data collected, including the questions, are found in a worksheet within the Teagle CS Senior Interview Results 2014, another link found in the appendix of this document.

Conclusions and Future Work

This past academic year I visited colleagues at Stanford University, a mecca for computing education, especially preparing for software development in the age of the startup. I described this project to each of them, and they all remarked at the lack of scale. I replied the same to each: “Exactly, this project is something Stanford cannot do but Haverford can.” A substantial amount of work goes into this type of assessment project, and it does not scale well, but we have found that this type of project helps to distinguish a degree in computer science from anywhere else.

Observations, both quantitative and qualitative, show that our students seem more comfortable discussing course concepts, that they mostly remember, in the abstract, as one would expect in an academic environment. And even though the interviews in the project were conducted by an academic, it became clear that the students were less comfortable with more practical matters

---

7 We also provide interactive grading, code reviews in the first course, and a supervised senior thesis, all labor intensive.
in software development. Fortunately, students are quite motivated to address this lack of preparation with software development, many asking us to conduct more interviews as practice (and they do not seem to have the same motivation for thesis presentation practice).

We in the Department plan to review the results (as soon as all reviews are in) and make curricular and other adjustments (e.g., professional presentations, hack-a-thons) to extend the preparation our students receive in computer science concepts to include software engineering experiences and principles. Our newest faculty member has introduced a course on mobile development which addresses some of these issues, and we will explore others as much as time, staffing and energy permit.

As our numbers continue to increase, we will have to adjust the methodology. We are considering partitioning all reviews among the four full-time faculty and inviting alumni to conduct and review seniors in this project\(^8\) (as well as our friends in OAR/CCP and the Library).

As required by the grant and throughout the process, the Teagle group shared our experience with Haverford colleagues. In year one, we produced video clips describing our projects. In year two, we hosted a lunch for interested faculty which highlighted the common senior year aspects of our projects. At the close of the grant, we will be circulating our reports more broadly to all faculty, inviting further conversation.

**Acknowledgements**

Cathy Fennell made my work possible by keeping me on track. The Department of Computer Science has been very supportive, and the students have been flexible in the scheduling and providing feedback to me about the project. Current Teagle project participants (i.e., Ana and Anne) at Haverford were very patient with my novel (i.e., crazy) idea; past Teagle people from the Tri-College were also helpful. The members of OAR/CCPA and the Library have provided either space, time to complete a rubric, or meet with me about this project (and completed their reviews before the CS faculty). Cris Fuller was helpful with the pilot, recording all interviews. The Office of the Provost has been supportive, thanks Fran and Kim. And again, Cathy really made the difference, keeping me on track, a great blend of pleasant persistence.

---

\(^8\) We invited these students, now almost all alumni, to mentor during the participant survey.
Appendix

Sample Interview Script:

The problem to solve involves feeding your kids -- McDonald’s sells McNuggets in groups of six, nine and fifteen. If you have a very large family, you need lots of nuggets -- and in little time, so an app would help. How would you go about designing a program to determine if we can (or cannot) buy sets of packages of McNuggets that adds up to a given number of McNuggets; better yet, if you cannot buy the sets/packages to give exactly the right number of McNuggets, how many packages should be bought to keep the kids happy? If the price per package is $4/6, $5.50/9 and $10/15, how can we feed the kids “best?” What if there is a deal, such as buy one 6-pack and get another free? Can this be generalized?

The key is this problem asks the student to put together an appropriate model using some combination of prose, math, graphs, etc. We can also see if they consider domain issues (e.g., number need to be non-negative integers), if the consider invariants and other reasoning tools. The first part is a simple math equation; namely, $6x + 9b + 15c = n$

But there are a few ways to solve this problem, including brute-force, stopping when a solution is found; “best” implies optimization, a more challenging problem. Finally, the open question “Can this be generalized?” can be interpreted as the interviewee deems, such as variable prices, more sets of McNuggets, even other food types.

It is key to appreciate that the problem has an accessible start, with deeper concepts needed to proceed; furthermore, it should be open as much as possible to however the interview unfolds to see how well the student can perform.

Other project resources:

- Rubric for Interview Assessment is available for download from this link.
  https://drive.google.com/file/d/0BwZWZYA_55txUV94ZXdTbmE4STQ/view?usp=sharing

- A brief Participant Survey, now closed, can be viewed at this link.
  http://goo.gl/forms/uwPfuBtsqZ

- A PDF of the survey as seen by the participants is available at this link.
  https://drive.google.com/file/d/0BwZWZYA_55txejJIUUFCUU4zaWc/view?usp=sharing

- All raw data (anonymized) is available within the Teagle CS Senior Interview Results 2014, found at this link.
  https://docs.google.com/spreadsheets/d/1hk51q5qum3Qzf9BW539JPoCZiRbERvyp9idRhsxw1eE/edit?usp=sharing
Student Learning Goals and Objectives

As Haverford faculty, our central goal is to propel each student to realize their full ability to think deeply and communicate clearly. Computer Science is our field of study, so we draw problems from this discipline to challenge and thereby strengthen thinking and communication skills.

These problems may focus on the foundations of our field (the study of the representation of information via concrete data structures and the manipulation of information via algorithms) or related areas of inquiry such as user interface design (combining the technical and psychological aspects of the relationship between people and computing systems), programming language design (the human element of the design of computing systems) or scientific computing (the use of computation to support other fields of scientific inquiry).

Algorithms and data structures are solutions to general problems of information processing or storage. The field of computer science abounds with situations in which a single problem can be solved in several ways, and with solutions that can be applied to a variety of problems. Advanced work in computer science requires the ability to identify and reason about problems, solutions, and the connection between the two.

Student learning goals are:

- [1] Each student will realize their full ability to think deeply. This goal involves mastering discipline-specific concepts such as abstraction, correctness and complexity, and recognizing their broad and deep applications, both theoretically and practically, in new contexts. Deep thinking also involves recognizing the difference between a problem, a solution, and a problem specification.

  Learning Objectives….Students will be able to:
  a Identify the role of abstraction in a computational problem situation; for example, distinguish a general problem from an specific instance, or understand the mapping between an abstract data type (ADT) and a given representation of that ADT
  b Develop original, correct solutions demonstrating an appropriate level of abstraction, using two or more design techniques specific to the field
  c Express a general solution in an appropriate programming language
  d Analyze and compare the efficiency of alternative solutions, both quantitatively and qualitatively
  e Increase the confidence in a solution by use various approaches, including proof, testing, and mathematical reasoning
f  Apply knowledge acquired in early courses to subsequent courses; this objective should occur from introductory to core courses, as well as from core to advanced electives and to the senior thesis

- **[2] Each student will communicate their thinking clearly and effectively.** This goal/objective involves taking a discovered or developed solution (or a given problem definition, etc ...) and sharing that solution with peers, managers, clients, and other professionals completely, persuasively and with appropriate use of vocabulary and other tools (e.g., charts, proofs, demonstrations).

  Learning Objectives...Students will be able to:
  a  Articulate their solution to others (e.g., peers, instructors, conference attendees), including why and how a solution solves a problem and what assumptions were made.  
  b  Use written documentation, models and examples to illustrate what counts as a solution to a given problem.
  c  Exhibit skills of the contemporary computing professional, including teamwork, persistence in product delivery, and “thinking on one’s feet”

- **[3] Each student will identify, interpret and evaluate the theoretical, practical and ethical implications of their work in the field.** This work is most easily identified as software, but other results might be papers written and published, projects chosen over others ignored, and even questions raised.

  Learning Objectives...Students will be able to:  
  a  articulate a broad perspective on the social and ethical implications of computing and information technology
  b  acquire specific knowledge about major issues in few distinct areas of the field of Computer Ethics (i.e., “breadth”).
  c  acquire in-depth knowledge of at least one significant ethical issue generated by information technology (i.e., “depth”).

---

1 adapted from *Computing Curricula 2013, Ironman draft, version 0.8*, November 2012, see http://ai.stanford.edu/users/sahami/CS2013//ironman-draft/cs2013-ironman-v0.8
# Computer Science Mock Interview Evaluation

**Interviewee:** ___________________________  **Date:** ________________

**Evaluated By:** ___________________________

## I. Content

<table>
<thead>
<tr>
<th>Task</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constructs specific instances when given a general problem [problem vs. problem instance]</td>
<td>4 3 2 1</td>
</tr>
<tr>
<td>Generalizes from a set of specific instances [problem vs. problem instance]</td>
<td>4 3 2 1</td>
</tr>
<tr>
<td>Proposes an appropriate model of the problem [general problem, problem vs. solution]</td>
<td>4 3 2 1</td>
</tr>
<tr>
<td>Develops appropriate algorithm and/or data representation when given an abstract specification [general problem, abstraction]</td>
<td>4 3 2 1</td>
</tr>
<tr>
<td>Discusses at the right level of abstraction, perhaps mapping from one to another level [abstraction]</td>
<td>4 3 2 1</td>
</tr>
<tr>
<td>Appreciates efficiency trade-offs [complexity]</td>
<td>4 3 2 1</td>
</tr>
<tr>
<td>Verbally critiques software components in terms of correctness and complexity [correctness, complexity]</td>
<td>4 3 2 1</td>
</tr>
<tr>
<td>Develops appropriate test cases, ranging from the conventional to the extremes [correctness]</td>
<td>4 3 2 1</td>
</tr>
<tr>
<td>Reasons about computation using appropriate rigorous techniques [correctness]</td>
<td>4 3 2 1</td>
</tr>
<tr>
<td><strong>Summary:</strong> Effectively summarizes the strengths/shortcomings of their chosen approach.</td>
<td>4 3 2 1</td>
</tr>
<tr>
<td>Other:</td>
<td>4 3 2 1</td>
</tr>
<tr>
<td>Other:</td>
<td>4 3 2 1</td>
</tr>
</tbody>
</table>

“Other” options might include consideration of ethical implications, appropriate self-correction of thinking/methodology, etc.

<table>
<thead>
<tr>
<th>Rating</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>Excellent</td>
</tr>
<tr>
<td>3</td>
<td>Good</td>
</tr>
<tr>
<td>2</td>
<td>Fair</td>
</tr>
<tr>
<td>1</td>
<td>Needs Attention</td>
</tr>
</tbody>
</table>

**Comments:**
# Computer Science Mock Interview Evaluation (Areas II & III)

**Interviewee:** ____________________________  
**Date:** _________________  
**Evaluated By:** ____________________________

## II. Connection (of content and communication)

<table>
<thead>
<tr>
<th>Description</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Correlation: Made effective reference to past/current course or project work.</td>
<td>4 3 2 1</td>
</tr>
<tr>
<td>Creativity: Adds to or expands on the original problem and their solution in an innovative or unexpected way.</td>
<td>4 3 2 1</td>
</tr>
<tr>
<td>Completeness: Responded directly to questions and offered full responses to questions.</td>
<td>4 3 2 1</td>
</tr>
</tbody>
</table>

## III. Communication (of content)

<table>
<thead>
<tr>
<th>Description</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Demeanor: Engaged, interested, sincere.</td>
<td>4 3 2 1</td>
</tr>
<tr>
<td>Enunciation: Words were enunciated clearly but without exaggeration.</td>
<td>4 3 2 1</td>
</tr>
<tr>
<td>Volume: Could be easily heard but was not shouting.</td>
<td>4 3 2 1</td>
</tr>
<tr>
<td>Pacing: Spoke slowly enough to be followed easily, but did not drag.</td>
<td>4 3 2 1</td>
</tr>
<tr>
<td>Expressiveness: Added interest by varying volume, pace, and pitch, or used gestures to enhance understanding.</td>
<td>4 3 2 1</td>
</tr>
<tr>
<td>Eye contact: Maintained appropriate eye contact with the camera and/or interviewer.</td>
<td>4 3 2 1</td>
</tr>
<tr>
<td>Use of Board/other media: Made effective and accurate use of the board to make thinking visible.</td>
<td>4 3 2 1</td>
</tr>
<tr>
<td>Legibility: Writing on the board or screen was clear, visible, and appropriately scaled so as to be visible on camera.</td>
<td>4 3 2 1</td>
</tr>
<tr>
<td>Other:</td>
<td>4 3 2 1</td>
</tr>
</tbody>
</table>

4 = Excellent  
3 = Good  
2 = Fair  
1 = Needs Attention

**Comments:**

---

[ NOTE: The image contains a table with a grid and text that is not fully visible due to cropping or other factors. The table includes columns for description and score, with ratings from 4 (Excellent) to 1 (Needs Attention). The text is focused on evaluating the connection and communication in a computer science mock interview context.]
CS Teagle Survey 2014

Please reply with your thoughts and recollections, and please try to reply by Oct 17 (late, sorry, I know, but the report is due Monday :) -- thanks, jd

* Required

Reactions and recollections *

Provide your response from the choices below.

<table>
<thead>
<tr>
<th>Strongly Disagree</th>
<th>Disagree</th>
<th>Neutral</th>
<th>Agree</th>
<th>Strongly Agree</th>
<th>Not Applicable</th>
</tr>
</thead>
<tbody>
<tr>
<td>I was not sure what JD was doing with this interview/assessment.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>The question was a good vehicle to demonstrate my capabilities in computing.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>The question was very different from ones I received in my other interviews.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>The time went faster than I expected during the interview.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I succeeded, and responded well in the interview.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I learned to appreciate various ways to solve</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
various ways to solve even a simple problem in computing.

I did not get to speak enough during the interview.

JD did not understand my response completely.

The interview helped was good practice for my other interviews.

The CS Lounge was not a great place for the interview.

From your experience, what topics/concepts from the first year of computer science courses (i.e., cs105 and cs106) do you use in your current job? *

What skills did your not get in you time at Haverford that you see other employees at the same level of experience already have? *

What is your current professional status? *
Click on all that apply.
- Industry - software development
- Industry - other (e.g., management, startup)
- Graduate Study
- Professional Study (e.g., law, medicine)
- Other: [ ]