Learning Goal to be Assessed:
Our Teagle Assessment project will focus on the oral component of our student learning goal on communication (goal 2 of the comprehensive departmental student learning goals, attached).

- [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.¹
  b Use written documentation, models and examples to illustrate what counts as a solution to a given problem.

While we expect our students to be able to take a given problem and develop an appropriate solution (that may involve an algorithm implemented as a program), our assessment project will target self-presentation effectiveness and “thinking on one’s feet” in a professional setting. The basic mechanism will be a “tech job interview.”

Learning Strategy:
The two-course introductory sequence in Computer Science (cs105 and cs106) provides both content knowledge and a requirement to communicate with peers and with instructors, both informally in lab and classroom settings, as well as more formally via code review and interactive grading and feedback sessions. Term projects are presented directly to the instructor for real-time feedback and grading. Those undertaking summer research participate in poster presentations. Foundational concepts are reinforced throughout the remainder of the core, elective, and thesis courses, so that students expand their knowledge base. Communication of deep thinking is practiced throughout, particularly in the Junior Seminar, even though each student will have a distinct path through the curriculum.

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
**Learning Assessment (measures):**
A scripted “Tech Job Interview” assessment will gauge both depth of understanding of computing problems/solutions and the ability to convey this understanding concisely and effectively. Field-specific skills such as appreciation for abstraction, awareness of options and model limitations, and thinking on one’s feet will be scored using a rubric reflective of desired outcomes within a professional job interview incorporating a “performance measure” (e.g. a problem-solving situation). These interviews will be video-recorded to facilitate scoring and to provide participants with formative assessment of their performance.

The faculty member in the Department who is the Teagle participant will administer these interviews in the fall of 2013 to:
- All CS majors (12 within the Class of 2014)
- A control group who have taken CS106, either Seniors (HC ‘14) who did not major in CS (Plan A--preferred) or a sample of the current enrollment in CS106 (Plan B--to note difference in performance without exposure to the complete CS curriculum)

The entire project will be piloted in the spring of 2013 with the 5 CS majors and hopefully a control group of 5 non-CS majors (seniors) who took the CS106 course.

A well-defined rubric for evaluation will help to maintain consistency and facilitate comparison. It will contain cover both discipline-specific content evaluation and assessment of personal presentation skills. The Teagle participant and at least one additional Computer Science faculty member will review the student videos and score performance, along with one staff member from the Office of Academic Resources and one staff member from the Career Development Office. Scoring might be done in a collective setting, with food served.

**Using the results to improve learning:**
The rubric scores will be tallied, analyzed and discussed within the department to inform subsequent planning for departmental curriculum, course content/delivery, and student evaluation mechanisms.

Additional analysis of participant information in relation to their performance scores will also be discussed by the Department. These data might include grade in CS106, completion of computing-related courses outside the department, overall GPA, majors/minors, foreign language background, and/or other student experiences/activities.
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
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”).

---

2 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