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:
- 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
- Use written documentation, models and examples to illustrate what counts as a solution to a given problem.
- 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: ²
- Articulate a broad perspective on the social and ethical implications of computing and information technology
- Acquire specific knowledge about major issues in few distinct areas of the field of Computer Ethics (i.e., “breadth”).
- 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