Notes for BASIC INFORMATION Cover Sheet

(*) Estimated enrollment for CSE 584 is 10–15.
Estimated enrollment for PHI 584 is 10–15.
Estimated enrollment for CSE/PHI 484 is 25.
Total estimated enrollment = 50.

(+) Prerequisite is graduate standing. Some familiarity with either computer science or philosophy would be helpful.

(**) Rank and degree of instructor: Associate Professor, Ph.D.; and Adjunct Professor of Philosophy

REQUIRED APPENDICES:

DESCRIPTION FOR CATALOG

An introduction to philosophical issues in computer science. Examines the nature of computer science (e.g., whether it is a science or a branch of engineering, whether it is the study of physical computers or abstract computing); the nature of computation, algorithms, and software, and their implementation in computer programs; the nature of computer programs (e.g., whether they are theories, models, or simulations; whether they can or should be patented or copyrighted; whether they can be verified). Also briefly explores the philosophy of artificial intelligence and issues in computer ethics.

COURSE OUTLINE

Course Objectives:

To introduce computer science students to some of the philosophical questions concerning their discipline, and to introduce philosophy students to some interesting new applications of philosophical concepts and methods of analysis.

Outline of Content:

What is philosophy? What is computer science? Is computer science a natural or artificial science of computers, the study of algorithms, a body of knowledge dealing with information-transforming processes, or a natural science of procedures? Or is it an engineering discipline (and what is the difference between science and engineering)? If it is a study of computers, then what is a computer? Is the brain a computer? If computer science is a study of algorithms, then what is an algorithm, what is computation, and can there be computable procedures that are not computable in Turing's sense? What is the relation of computation to the real world? what is the relation of software to hardware? What is implementation? Are computer programs scientific theories? Can they be patented? Or should they be copyrighted? Can they be verified? Could we build artificial intelligences? Should we? Are there decisions computers should never make?

Course Bibliography:

What is philosophy?

Read at least one of the following:

1. Woodhouse, Mark B. (2003), *A Preface to Philosophy* (Wadsworth), Chs. I–III (pp. 1–45).
2. Colburn, Timothy R. (2000), *Philosophy and Computer Science* (M.E. Sharpe), Chs. 3–4 (pp. 19–50).
3. Plato, *The Apology* (various versions on line)
4. Audi, Robert (2001), "Philosophy: A Brief Guide for Undergraduates" (on line)

What is computer science?

1. Newell, Allen; Perlis, Alan J.; & Simon, Herbert A. (1967), "Computer Science", *Science* 157: 1373–1374.
2. Knuth, Donald (1974), "Computer Science and Its Relation to Mathematics", *American Mathematical Monthly* 81: 323–343, §1.
3. Newell, Allen, & Simon, Herbert A. (1976), "Computer Science as Empirical Inquiry: Symbols and Search", *Communications of the ACM* 19: 113–126, pp. 113–116, 120, & "Conclusion" (pp. 125–126).
4. Denning, Peter J.; Comer, Douglas E.; Gries, David; Mulder, Michael C.; Tucker, Allen; Turner, A. Joe; & Young, Paul R. (1989), "Computing as a Discipline", *Communications of the ACM* 32: 9–23, pp. 9–12, 16ff.
5. Hartmanis, Juris, & Lin, Herbert (eds.) (1992), "What Is Computer Science and Engineering?", in Juris Hartmanis & Herbert Lin (eds.), *Computing the Future: A Broader Agenda for Computer Science and Engineering* (National Academy Press): "Computer Science & Engineering" (pp. 163–168) and "Abstractions in Computer Systems" (pp. 168–174).
6. Brooks, Frederick P., Jr. (1996), "The Computer Scientist as Toolsmith II", *Communications of the ACM* 39: 61–64.
7. Shapiro, Stuart C. (2001), "Computer Science: The Study of Procedures" (on line)

What is science? What is engineering?

1. Papineau, David (1996), "Philosophy of Science", in Nicholas Bunnin & E.P. Tsui-James (eds.), *The Blackwell Companion to Philosophy* (Blackwell), esp. pp. 290–294, 298–308, 319–320.
2. Kemeny, John G. (1959), *A Philosopher Looks at Science* (Van Nostrand): Intro., Ch.5 ("The Method"), Ch.10 ("What Is Science?")
3. Davis, Michael (1998), *Thinking Like an Engineer: Studies in the Ethics of a Profession* (Oxford): Ch.1 (pp. 3–17), pp. 25–28, pp. 31–37.

What is a computer?

1. "A Very Brief History of Computers" (on line)
2. O'Connor, J.J., & Robertson, E.F. (1998), "Charles Babbage" (on line)
3. Simon, Herbert A., & Newell, Allen (1958), "Heuristic Problem Solving: The Next Advance in Operations Research", *Operations Research* 6: pp. 1-3 (on Babbage).
4. Ensmenger, Nathan (2004), "Bits of History: Review of A.R. Burks's Who Invented the Computer? The Legal Battle that Changed Computing History", *American Scientist* 91 (September-October): 467–468.
5. Searle, John R. (1990), "Is the Brain a Digital Computer?", *Proceedings and Addresses of the American Philosophical Association* 64: 21–37.
6. Hayes, Patrick J. (1997), "What Is a Computer? An Electronic Discussion", *The Monist* 80(3) (on line).

What is an algorithm?

1. Henkin, Leon (1962), "Are Logic and Mathematics Identical?", *Science* 138: 788-794.
2. Herman, Gabor T. (1983), "Algorithms, Theory of", in Anthony S. Ralston (ed.), *Encyclopedia of Computer Science and Engineering* (Van Nostrand Reinhold): 57–59.
3. Turing, Alan M. (1936), "On Computable Numbers, with an Application to the *Entscheidungsproblem*", *Proceedings of the London Mathematical Society* Ser. 2, Vol. 42: 230–265.
4. Soare, Robert I. (1996), "Computability and Recursion", *Bulletin of Symbolic Logic* 2: 284-321.

What is a procedure?

1. Preston, Beth (2000), "Recipes and Songs: Towards a Theory of Production" (on line) (skim §1-2; read §§3-4)
2. Cleland, Carol E. (1993), "Is the Church-Turing Thesis True?", *Minds and Machines* 3: 283–312.

What is hypercomputation?

1. Copeland, B. Jack (2002), "Hypercomputation", *Minds and Machines* 12: 461–502.
2. Kugel, Peter (2002), "Computing Machines Can't Be Intelligent (... and Turing Said So)", *Minds and Machines* 12: 563–579.

What is a computer program/What is implementation?

1. Chalmers, David J. (1993), "Does a Rock Implement Every Finite-State Automaton?", *Synthese* 108 (1996): 309–333.
2. Rapaport, William J. (1999), "Implementation Is Semantic Interpretation", *The Monist* 82: 109–130.

What is a computer program/Are programs theories?

1. Weizenbaum, Joseph (1976), *Computer Power and Human Reason: From Judgment to Calculation* (Freeman): Ch.5 ("Theories and Models"), pp. 132–153; Ch.6 ("Computer Models in Psychology"), pp. 154–181.
2. Simon, Herbert A. (1996), *The Sciences of the Artificial* (MIT), Ch.1 ("Understanding the Natural and Artificial Worlds"), pp. 1–24.

What is software?

- Moor, James H. (1978), "Three Myths of Computer Science", *British Journal for the Philosophy of Science* 29: 213–222.

Can programs be copyrighted or patented?

- Newell, Allen (1985-1986), "Response: The Models Are Broken, the Models Are Broken", *University of Pittsburgh Law Review* 47: 1023–1031.

Can programs be verified?

- Smith, Brian Cantwell (1985), "Limits of Correctness in Computers", *Technical Report CSLI-85-36* (Center for the Study of Language and Information); first published in Charles Dunlop & Rob Kling (eds.), *Computerization and Controversy* (Academic, 1991): 632–646.

Philosophy of AI

1. Turing, Alan M. (1950), "Computing Machinery and Intelligence", *Mind* 59: 433–460.
2. Searle, John R. (1980), "Minds, Brains, and Programs", *Behavioral and Brain Sciences* 3: 417–457.

Computer Ethics

1. Moor, James H. (1979), "Are There Decisions Computers Should Never Make?", *Nature and System* 1: 217–229.
2. LaChat, Michael R. (1986), "Artificial Intelligence and Ethics: An Exercise in the Moral Imagination", *AI Magazine* 7(2): 70–79.

Philosophy of Computer Science

1. Scheutz, Matthias (2002), "Philosophical Issues about Computation", *Encyclopedia of Cognitive Science* (Macmillan) (on line)
2. Smith, Brian Cantwell (2002), "The Foundations of Computing", in Scheutz, Matthias (ed.), *Computationalism: New Directions* (MIT): 23–58.

Potential Texts:

There are no ideal texts for this course, hence the large number of readings to be made available to students on line.

Plans for Course Evaluation:

Both a midsemester and an end-of-semester course evaluation will be given: The former is for the instructor's benefit while the course is in progress; the latter is mandated by SEAS. During the trial version, there was nearly unanimous praise from the students for the content and format of the course. Here is one unsolicited email message sent by a student after the course was over:

> I'd like to thank you for putting together such a great course this semester. I'll admit, knowing very little about it, I never had much respect for philosophy in the past- but this course has provided me with an entirely new perspective. In fact, I'd say that I learned as much in your course as any other I've taken in my graduate career at UB (not to mention the fact that the skills I learned in 510 are far more transferable than the skills of the more esoteric CS courses).
>
> I noticed that 510 is officially reserved for "special topics", so I'm not sure what your plans are- but I urge to offer this course again in the future. It offers exactly the kind of breadth of education that the department needs to stress, and with its CS flavor, it can tap the interest of students who would otherwise blow it off.
>
> Thanks again for a great semester, and please consider making Philosophy of CS a regular offering :)

Grading:

Students will be required to write approximately 5 1-page position papers (to be peer-reviewed and revised), and either take a final exam or write a term paper. Letter grades will be based approximately 50% on written work and 50% on class attendance and participation.

RATIONALE

There are many courses offered by philosophy or computer science departments on the philosophy of artificial intelligence, but I have been able to find only two courses (both in Europe) on the broader topic of the philosophy of computer science.

There are many substantive questions about the nature of computer science as an academic discipline, the nature of computation, the nature of physical computing devices, the relationships between abstract computation and concrete computing, etc., that are not covered either in typical philosophy courses or, certainly, in typical computer science courses.

In addition, a synoptic view of all of these issues can serve as an excellent introduction to, or as a capstone for, a major in computer science.

The Spring 2004 trial version of the course had two additional features that seem to warrant making this a permanent addition to our course offerings:

1. It was popular (enrollment of just under 50 students).
2. The students found it valuable, not only for its coverage of topics in the philosophy of computer science, but also for the critical-thinking skills they learned.

DUPLICATION CHECK

The only other course that is remotely similar is the Department of Philosophy's PHI 398: Philosophy of Artificial Intelligence. The proposed course differs from this one in the following ways:

1. This is a graduate course, whereas PHI 398 is an undergraduate course.
2. The philosophy of artificial intelligence (AI) is a subdiscipline of the philosophy of computer science. Thus, the proposed course explores issues in the philosophy of AI as only one topic, covering only about 1 week's worth of material.

CROSSLISTING

CSE 584 will be crosslisted with PHI 584. This will enable graduate students in PHI to take the course for PHI credit.

CSE 584 will also be dual listed with CSE 484 (and with PHI 484). Where the 500-level version might serve as an excellent overview course for graduate students, the CSE 400-level version might serve as an excellent capstone course for advanced undergraduates. Graduate students will be required to write several essays that will be merely optional for undergraduates, and graduate students will be graded on a different scheme from undergraduates.