5

The Impact of Computers: A Syllabus

Edward T. Ordman

MEMPHIS State University, drawing on the resources of large undergraduate and graduate programs in computer science, has established a survey course on the impact of computers for upper-level nontechnical students. The syllabus and reading list for this course may suggest topics for reading or in-service courses for teachers that are different from the relatively common "introduction to programming."

It is increasingly accepted that all teachers, perhaps all college graduates, and possibly even all high school graduates, should have some acquaintance with computers. Although a brief introduction to computer programming is often proposed, many such introductions fail to address the issue of how computers actually affect society or specific institutions.

During the early 1980s, Memphis State University's Department of Mathematical Sciences developed a computer literacy course intended for nontechnical students at the senior or graduate level. This course was intended, not as a programming course, but as a broad overview of computer science, computer applications, and the impact of computers on organizations and society.

The project reported here was a joint activity with Charles Brandon, Stanley Franklin, Hugh McHenry, and Austin Smith. It was supported by Memphis State University and other donors. Staff for the first offerings of the course included the following persons:

- A successful corporate executive with broad experience in computer consulting and directing corporate computer operations
- The chairman of a university computer science department (Memphis State)
- An experienced teacher with an extensive background in computer-aided and computer-managed instruction
- An experienced computer science teacher with government and consulting experience, involved in recent years in building a microcomputer-based computer science program

- A computer consultant with extensive knowledge of computer hardware and medical and library applications

The development of the course was undertaken with the specific expectation that there would be spin-offs in the form of shorter courses, particularly extension courses and teacher workshops. Accordingly, the detailed syllabus for this course may suggest appropriate topics or groups of topics for in-service workshops or individual reading.

The syllabus

The general breakdown of the lectures was as follows:

Topic	Hours	Percent
Hardware	3	7.5
Systems programs	4	10
Introduction to BASIC	4	10
Theoretical computer science	3	7.5
Applications	8	20
Impact on organizations	8	20
Impact on society	9	22.5
Miscellaneous	1	2.5
	40	100

The calendar

Although the calendar naturally varied somewhat from one semester to another, the following was typical:

Week 1. Introduction; Boolean algebra; simple circuits, gates, flip-flops, and adders. *Laboratory:* Use of digital trainers to build flip-flops, half-adders, and simpler circuits.

Week 2. History of computers from 1890 to 1960; a look at artifacts (old core memories, plug-board wiring, modern circuit chips, disks, etc.); brief introduction to machine language. What is an assembler? A compiler? An operating system? *Laboratory:* Introduction to the use of an IBM personal computer; game playing.

Week 3. More about compilers and operating systems; interpreters; first lecture about programming in BASIC. *Laboratory:* Start of programming in BASIC (mainly typing in commands from an instruction sheet to see the result).

Week 4. More lectures on BASIC (mainly built around writing a program to balance a checkbook). Second BASIC laboratory.

Week 5. Examples of scientific programming; discussion of simulating aircraft operation, atmospheric modeling; the concept of structured

programming; noting other programming languages. Third BASIC laboratory.

Week 6. Extensive example—how does a business (e.g., a law firm) decide it needs a computer? How does it decide what to buy? What is word processing? How does it relate to data processing? *Laboratory:* Use of a word-processing program.

Week 7. What is artificial intelligence? How do very large programs get written? Top-down programming; management of programming. Second word-processing laboratory.

Week 8. Data base management systems; distributed data bases; computer networking. *Laboratory:* Use of a (small, local) data base management system.

Week 9. Electronic spreadsheet programs for business calculations and forecasting. *Laboratory:* Use of VISICALC.

Week 10. Impact of computers on a business organization; economics of the organization; funding the data-processing operation; internal security of data and equipment. Second electronic spreadsheet laboratory.

Week 11. Life cycle of data processing in the corporation; impact on the organizational structure. *Laboratory:* Use of a personal computer as a terminal to access an on-line, public data bank.

Week 12. Impact of computers on society: new tools, new careers, new businesses. *Laboratory:* Further use of on-line data banks.

Week 13. Computer impacts: privacy problems, computer crime, transborder data flow, displacement of workers. *Laboratory:* Open for work on student projects.

Week 14. Cultural and artistic impact of computers; computer-aided instruction; psychological impact of computers. *Laboratory:* Open for work on student projects.

Credit arrangements

Four semester hours of credit were given for the course, which met for three hours of lecture and three hours of laboratory weekly for fourteen weeks. A brief term paper (at least five pages, prepared on a word processor and requiring output from at least one other program) was also required. The laboratory was available to students for about six hours a week in addition to the scheduled three hours, so that they could use the computers, review slide sets shown in class, or consult with the laboratory assistants. Students enrolled in certain career-oriented programs had the option of undertaking substantially larger projects to bring the total credit up to six hours; some of these projects resulted in significant contributions to the

resources available for the course. (One student, through his employer, had access to a Boeing 747 flight simulator!)

Reading list

The textbook (in which we jumped around considerably) was *Computers in Society*, 3d edition, by Donald H. Sanders (New York: McGraw-Hill, 1981). We also used just over half the slide sets included in "The Audio Visual Library of Computer Education" (Prismatron Productions, Inc., Mill Valley, Calif.).

During the development of the course the instructors collected a list of books that could appropriately be read to supplement the text. It included many books that we felt could be read by someone without an extensive technical background but that still had substantial content for the material we were studying. We were less successful than we hoped in getting the students to read extensively in these books but found them very useful in preparing our own lectures.

James L. Adams, *Conceptual Blockbusting* (New York: W. W. Norton & Co., 1980)

Karen Billings and David Moursund, *Are You Computer Literate?* (Forest Grove, Oreg.: Dilithium Press, 1979)

Frederick P. Brooks, Jr., *The Mythical Man-Month: Essays on Software Engineering* (Reading, Mass.: Addison-Wesley Publishing Co., 1975)

H. Dominic Covvey and Neil Harding McAlister, *Computer Consciousness* (Reading, Mass.: Addison-Wesley Publishing Co., 1980)

Tom de Marco, *Concise Notes on Software Engineering* (New York: Yourdon, 1979)

Engineering Concepts Curriculum Project, *Man and His Technology* (New York: McGraw-Hill Book Co., 1973)

Robert Fenichel and Joseph Weizenbaum, Introduction to *Readings from Scientific American: Computers and Computation* (San Francisco: W. H. Freeman & Co., 1971)

Louis E. Frenzel, Jr., *The Howard W. Sams Crash Course in Microcomputers* (Indianapolis: Howard W. Sams & Co., 1980)

Herman R. Goldstine, *The Computer from Pascal to Von Neumann* (Princeton, N.J.: Princeton University Press, 1972)

Frank Herbert with Max Barnard, *Without Me You're Nothing: The Essential Guide to Home Computers* (New York: Simon & Schuster, 1980)

Martin O. Holoien, *Computers and Their Social Impact* (New York: John Wiley & Sons, 1977)

Tracy Kidder, *The Soul of a New Machine* (Boston: Little, Brown & Co., 1981)

Don Lancaster, *The Incredible Secret Money Machine* (Indianapolis: Howard W. Sams & Co., 1978)

Murray Laver, *Computers and Social Change* (New York: Cambridge University Press, 1980)

Blaise W. Liffick, ed., *Program Design* (Peterborough, N.H.: Byte Books, 1978)

James Martin, *The Wired Society* (Englewood Cliffs, N.J.: Prentice-Hall, 1978)

Pamela McCorduck, *Machines Who Think* (San Francisco: W. H. Freeman & Co., 1979)

Glenford J. Myers, *Reliable Software through Composite Design* (New York: Petrocelli/Charter, 1975)

John Nevison, *Executive Computing* (Reading, Mass.: Addison-Wesley Publishing Co., 1980)

———, *The Little Rock of Basic Style* (Reading, Mass.: Addison-Wesley Publishing Co., 1978)

Adam Osborne, *Running Wild: The Next Industrial Revolution* (Berkeley, Calif.: Osborne/McGraw Hill, 1979)

Seymour Papert, *Mindstorms: Children, Computers, and Powerful Ideas* (New York: Basic Books, 1980)

George Penney, *Managing Computers: Data Processing Case Histories* (Rochelle Park, N.J.: Hayden Book Co., 1975)

Robert Persig, *Zen and the Art of Motorcycle Maintenance* (New York: Bantam Books, 1974)

Donald Sanders and Stanley Birkin, *Computers and Management in a Changing Society* (New York: McGraw-Hill Book Co., 1980)

Scientific American, *Microelectronics* (San Francisco: W. H. Freeman & Co., 1977)

Carl Townsend and Merle Miller, *How to Make Money with Your Microcomputer* (Forest Grove, Oreg.: Dilithium Press, 1981)

Dennie Van Tassel, *Program Style, Design, Efficiency, Debugging, and Testing* (Englewood Cliffs, N.J.: Prentice-Hall, 1978)

Mitchell Waite and Michael Pardee, *Microcomputer Primer* (Indianapolis: Howard W. Sams & Co., 1980)

Carl Warren and Merle Miller, *From the Counter to the Bottom Line* (Forest Grove, Oreg.: Dilithium Press, 1979)

Gerald M. Weinberg, *The Psychology of Computer Programming* (New York: Van Nostrand Reinhold Co., 1971)

Thomas Whiteside, *Computer Capers: Tales of Electronic Thievery, Embezzlement, and Fraud* (New York: New American Library, Mentor Books, 1978)

Jerry Willis, *The Peanut Butter and Jelly Guide to Computers* (Forest Grove, Oreg.: Dilithium Press, 1978)

———, *The Techno/Peasant Survival Manual* (New York: Bantam Books, 1980)

Edward Yourdon, *Managing the Structured Techniques* (Englewood Cliffs, N.J.: Prentice-Hall, 1979)

———, *Techniques of Program Structure and Design* (Englewood Cliffs, N.J.: Prentice-Hall, 1975)

Rodnay Zaks, *DON'T (or How to Care for Your Computer)* (Berkeley, Calif.: Sybex, 1981)