**IT literacy – evolution, curriculum and a modular e-text**

IT literacy courses date back to the 1960s at Dartmouth College. The IT literacy curriculum has been revised over the years as new platforms for developing and delivering applications emerged – time-sharing systems, personal computers and now the Internet. I am developing a set of IT teaching modules for the Internet era.

I will review the evolution of the IT literacy course and describe my Internet era curriculum and the modular courseware I am developing to teach it.

**Evolution of IT literacy**

John Kemeny and Thomas Kurtz offered the first IT literacy course at Dartmouth College in the early 1960s. They believed that every liberally educated undergraduate needed knowledge of computers and information technology, regardless of their major (Kemeny and Kurtz, 1968). The US National Science Foundation agreed, and funded their work (Kemeny and Kurtz, 1967).

Let’s look at the way IT literacy has evolved over time.

There have been four major platform innovations since the invention of the digital computer: batch processing, timesharing, the personal computer and the Internet. (The mobile Internet may prove to be the next major platform for delivering IT applications).

During the batch processing era, there were no IT literacy courses. Users submitted jobs on punch cards to an operator who grouped them with other jobs, and ran them through the computer in batches. The user picked up printouts and other output as much as a day later. People recognized that computers were significant technology at the time, but an IT literacy course would have been impractical.

With the advent of timesharing in the early 1960s, users began interacting directly with computers (Licklider, 1960). Kemeny and Kurtz realized that an IT literacy course was feasible, but there was little infrastructure to support it. There were experimental time sharing systems at a few universities,[[1]](#footnote-1) but none commercially available, so they created their own – the Dartmouth Time Sharing System, (Dartmouth College).

Kemeny and Kurtz felt that an introduction to algorithmic thinking and problem solving should be part of the curriculum, so they also invented a very simple programming language, BASIC, which was intended for instruction only, not practical applications.[[2]](#footnote-2)

With this infrastructure available, they began teaching IT literacy. The goal of their course was to teach the *skills and concepts needed for success as a student and after graduation as a professional and a citizen*. Those skills and concepts have changed over time as the technology and IT platforms have evolved, but this remains the goal of my course.

A high-level outline of the timesharing era IT literacy curriculum was:

* Concepts
  + Applications
  + Technology
  + Implications
* Skills
  + Simple programming
  + Algorithmic thinking

This curriculum was widely taught during the 1960s and 1970s, until the personal computer emerged as a significant platform for developing and delivering IT applications. The concepts portion of the course remained fairly stable, but the skills we covered changed. The emphasis on programming and algorithmic thinking was significantly reduced or even eliminated, and replaced with training in the use of personal computer productivity applications. The curriculum shifted to:

* Concepts
  + Applications
  + Technology
  + Implications
* Skills
  + Simple programming
  + Algorithmic thinking
  + Word processing
  + Spreadsheets

With the rise to market dominance of Microsoft Windows and Microsoft’s Office applications, programming and algorithmic thinking were squeezed out and the curriculum became:

* Concepts
  + Applications
  + Technology
  + Implications
* Skills
  + Word
  + Excel
  + PowerPoint
  + Access

This curriculum is still widely taught today. For example, two departments on my campus currently offer it. However, it is out of date. Today’s students need concepts and skills for the Internet era, which I address in the modular course I am developing. I will outline my course, and then discuss the modular approach I have taken to its implementation.

**IT literacy for the Internet era[[3]](#footnote-3)**

The following is the top level outline of my IT literacy course for the Internet era:

* Concepts
  + Applications
  + Technology
  + Implications
* Skills
  + Application development
  + Content creation
  + User skills

Let’s discuss the concepts first, then the skills. While the applications, implications and technology have changed markedly over the years, these categories remain valid.

The applications, the things we use information technology for, have exploded during the Internet era and are far different than they were in the timesharing era. My goal in this portion of the course is to introduce the students to a variety of the uses for information technology. The focus is on network-based applications for productivity and collaboration and their characteristics.

Students need a simple mental model of the way technology works in order to be effective users who can learn on their own and communicate with technicians, and, perhaps most important, to avoid a feeling of passive alienation, a fear of the technology they are using (Illich, 1973). As such, I present an overview of communication, storage and processing technology. Organizing around these functions remains valid even as the physical technology changes. We stress function, while pointing out that specific technologies are transient – improving steadily until they are displaced. This leads to a natural understanding of trends.

Of course some technical concepts are invariant. For example measures of data quantity and transmission rate and coding schemes that enable the representation of varying data types are as valid today as they were in the batch processing era.

I also cover the implications of IT for individuals, organizations and society. The early IT literacy courses had some coverage of social implications, but IT had little impact on the structure and role of organizations and the lives of individuals. That is clearly no longer the case, and current and future implications are a significant part of the course.

Limited critical thinking also falls into this portion of the course. Some consider critical thinking to be a major topic in an IT literacy course. For example, Cengage defines critical thinking ability as "a set of higher-order thinking and reasoning skills required for understanding and solving problems as they arise in modern technological systems.” (Cengage)

While these are important skills, they are beyond the scope of the IT literacy course as I conceive it. We already have many skills and concepts to cover, and higher order thinking and reasoning are better dispersed throughout the university and illustrated in context in other courses. However, we do cover a subset of critical thinking, *skeptical thinking –* the ability to discover and critically evaluate the logical consistency and provenance of information on the Internet.

While the top level organization of the concepts portion of the course has remained constant over time, the skills have completely changed.

The skills taught today typically center on Microsoft Windows and Office applications, but I find that today’s students are able to use Windows, do basic word processing, and create simple presentations and spreadsheets when they come to my class. I believe that that is sufficient for IT literacy – we are interested in the forest in this class, not the trees. More specialized skills with productivity software should be taught in context in more specialized classes. For example, advanced spreadsheet techniques can be introduced when needed in a finance or engineering class. An IT literate person will also be required to acquire new skills throughout their career.

Rather than spend time on skills with desktop productivity applications, I cover application development, content creation and to a lesser extent user skills with existing programs. Let’s discuss each area, beginning with application development.

During the batch processing and time sharing eras, only trained programmers developed applications. With the advent of the personal computer, end users began developing some of their own applications. For example, they used spreadsheets and simple data bases for forecasting and record keeping, word processing and page layout programs for documents and newsletters, and presentation software for work that had previously been done by graphics professionals.

The Internet is now a primary platform for developing and delivering applications, and doing so is cheaper and easier than in the past. Professionals are still needed to develop Internet services and major systems, but the process is greatly facilitated by access to the application program interfaces of programs and databases and the scalability of cloud infrastructure.

IT literacy students are not professional developers, but they are able to develop powerful applications using generalized Internet services. These services let us introduce application development on wikis, blogs, threaded discussions, databases, and so forth in this portion of the course.

Today’s students also need to be able to create text, image, audio and video content. I assume they are familiar with the mechanics of text processing (simple word processing) and focus on techniques for Internet writing – writing short documents, conversational documents and collaborative documents. Since their skills with the other data types are generally limited, I introduce the mechanics of capturing, editing, compressing and publishing image, sound and video data.

User skill with existing programs is of lesser importance than content creation and application development. This portion of the course is concerned with refining skills with applications they already use, for example talking about search options and strategy or the use of tags for information organization and retrieval.

In summary, the following is an annotated outline of my current course:

* Concepts
  + Applications (characteristics of network-based applications)
  + Technology (processing, storage and communication)
  + Implications (for individuals, organizations and society)
* Skills
  + Application development (using generalized Internet services)
  + Content creation (text, image, audio and video)
  + User skills (with existing network-based applications)

**IT literacy courseware**

I am creating a modular electronic text for this curriculum. I currently have 104 modules, which are roughly equivalent in scope to textbook sections -- we cover around 6 modules per week. Each module is focused on a few related skills or concepts, follows a standard format and has a unique URL.

The modules contain concise PowerPoint presentations with notes for each slide, narrated videos of the presentations using the notes as a script, illustrated transcripts, links to external resources, self-study questions and assignments.[[4]](#footnote-4) The modules are easily extensible, For example, one could simply add a link to an executable demonstration or simulation program if relevant.

I use the annotated PowerPoint presentations in the classroom. The narrated videos are for self- study by students who prefer images with audio and the illustrated transcript is for students who prefer images and text.[[5]](#footnote-5) This module format is consistent with both Mayer’s multimedia learning principles (Mayer, 2009) and the recognition of individual differences in preferred learning modes (Jonassen and Grabowski, 1993).

As noted, this courseware is modular.[[6]](#footnote-6) I have long advocated a modular approach to courseware (Press, 1994), and, while full-course print and electronic textbooks are dominant today, modular content is available at repositories like Merlot[[7]](#footnote-7) and the Kahn Academy.[[8]](#footnote-8) McGraw Hill allows a professor to assemble selected textbook sections, case studies and articles into a single text (online or print), but they are restricted to McGraw Hill material or the professor's own material.[[9]](#footnote-9) Nature Publishing has announced what appears to be an important series of highly customizable digital textbooks, the first of which recently became available (Press, 2011).

A modular approach is more versatile than a monolithic text. I am building enough modules for two overlapping courses – IT literacy for IT majors and IT literacy for non-majors. I teach a full course using selected modules, but Individual modules can also be used as supplementary material in other courses. This allows a professor to customize the content for his or her student’s goals and backgrounds.

A modular approach has a second major advantage – it lends itself to community development and collaboration. Our modular implementation meets Banker’s criteria for success as a commons-based peer production project: non-monetary rewards, low marginal distribution cost, and production in discreet, multi-size units that can be integrated at low or no cost, eliminating the need for hierarchical management (Benkler, 2002).

As with a Wikipedia article, students and teachers using a module can collaborate with its creator to improve it. For example, a student could leave a comment on an assignment explaining his or her solution or a professor could leave a comment on a topic module suggesting a new example or way to explain a concept. Students and faculty members can also create new modules themselves.

I implemented the modules using simple, free Web services. I use two blogs – one for the topic module (PowerPoint presentations, transcripts and other teaching material) and a second for assignments. Topic modules link to the corresponding assignments and the assignments link back to the corresponding topic module. Since every topic module and assignment has a unique URL, the weekly course schedule of topic modules and assignments can be a simple spreadsheet or wiki page. The appendix lists addresses of the online material.

**Conclusion**

I outlined the evolution of the IT literacy curriculum from its inception on time shared computers in the 1960s, and I described my Internet era IT literacy curriculum and its modular implementation.

Kemeny and Kurtz were prescient in understanding that a liberally educated person needed a basic understanding of information technology as much as they needed a basic understanding of math or literature – that IT literacy was for everyone, not just engineering, science and business students. Time has proved them to have been correct, but perhaps too conservative because information technology has grown increasingly important to us as individuals and in our organizations and society.

Kemeny and Kurtz could not have envisioned texting, Facebook and Internet dating, virtual organizations and telecommuting or the Arab Spring. As information technology has become more pervasive and more central to our lives and institutions, the IT literacy course has expanded. The curriculum I have outlined overflows a single course.

I believe it is time for us to require two IT literacy courses in the general education curriculum, just as we require multiple courses in mathematics, English, science and the humanities. Good luck getting that past the General Studies Committee on your campus.

**Appendix: Further information on the electronic text:**

* About the electronic text: http://bit.ly/fSxEYP
* About the course: http://bit.ly/h9MpIr
* This semester’s class (in progress): http://bit.ly/plXTXQ
* This semester’s topic modules (in progress): http://bit.ly/nN4KSS
* This semester’s assignments (in progress): http://bit.ly/nu0oRB
* Blog with current events relevant to the class: http://bit.ly/qEGci2
* Twitter feed with current events relevant to the class: http://bit.ly/qt3Ng5
* Google+ stream with current events relevant to the class: http://bit.ly/pSzeBa

**References**

Benkler, Yochai. Coase’s Penguin: Linux and the Nature of the Firm, Yale Law Journal, December 2002, http://www.yale.edu/yalelj/112/BenklerWEB.pdf.

Cengage Learning, Critical Thinking, accessed October, 2011, <http://www.cengagesites.com/academic/assets/sites/5544/DigitalLiteracyWhitepaper.pdf>.

Dartmouth College, History and description of the Dartmouth timesharing project, accessed October 2011, <http://dtss.dartmouth.edu/history.php>.

Frydenberg, Mark and Press, Larry, From Computer Literacy to Web 2.0 Literacy: Teaching and Learning Information Technology Concepts Using Web 2.0 Tools. Information Systems Education Journal, 8 (10). ISSN: 1545-679X, April 2010, <http://isedj.org/8/10/index.html>.

Illich, Ivan, *Tools for Conviviality.* New York: Harper and Row, 1973.

Jonassen D. and Grabowski, J. P., Handbook of Individual Differences, Learning and Instruction, Erlbaum, Hillsdale, NJ, 1993.

Kemeny, John G. and Thomas E. Kurtz, The Dartmouth Time-Sharing Computing System, Final Report to the National Science Foundation), June 1967, <http://www.eric.ed.gov/PDFS/ED024602.pdf>.

Kemeny, John G., and Kurtz, T. E., Dartmouth Time-Sharing, Science, Vol. 162, No 3850, October 11, 1968, pp. 223-228, <http://dtss.dartmouth.edu/sciencearticle/>. The article describes the project, including the rationale for computing being part of the liberal arts and the use of the system in and out of the curriculum.

Licklider, J. C. R., Man-Computer Symbiosis, IRE Transactions on Human Factors in Electronics, March, 1960, 4-11, http://som.csudh.edu/fac/lpress/lickliderMan-computer symbiosys.pdf.

Mayer, Richard E., Multimedia Learning, Cambridge University Press, 2009.

Press, Larry, Tomorrow's Campus -- Information Processing at Tomorrow's University, Communications of the ACM, Vol 37, No 7, pp 13-17, July, 1994.

Press, Larry, Textbooks – moving beyond the “Gutenberg Bible,” May 2011, <http://cis471.blogspot.com/2011/05/textbooks-moving-beyond-gutenberg-bible.html>.

1. CTSS at MIT was the most noted and influential, <http://en.wikipedia.org/wiki/Compatible_Time-Sharing_System> and <http://larch-www.lcs.mit.edu:8001/~corbato/sjcc62/>. [↑](#footnote-ref-1)
2. <http://en.wikipedia.org/wiki/Dartmouth_BASIC>. [↑](#footnote-ref-2)
3. This curriculum was worked out on a blog between April 2008 and June 2010, <http://computerliteracy3.blogspot.com/>, which influenced the start of my course and one offered at Bentley University (Frydenberg, Mark and Press, 2010). [↑](#footnote-ref-3)
4. Since this e-text is a work in process, some modules will be incomplete at any given time. However, all have at least the annotated PowerPoint presentation with self-study questions and links to external resources and corresponding assignments. [↑](#footnote-ref-4)
5. In practice, I only deliver some of the presentations that I assign in class. The students are expected to study the others on their own, leaving us time for discussion and clarification. [↑](#footnote-ref-5)
6. I also created a modular introduction to programming several years ago, <http://som.csudh.edu/fac/lpress/vbmodules/>. [↑](#footnote-ref-6)
7. http://www.merlot.org/merlot/index.htm. [↑](#footnote-ref-7)
8. http://www.khanacademy.org/. [↑](#footnote-ref-8)
9. http://create.mcgraw-hill.com/. [↑](#footnote-ref-9)