R e p o r t

History in the Computing Curriculum

I F I P

TC 3 / TC 9
Joint Task Group

John Impagliazzo
(Task Group Chair)
Hofstra University

Martin Campbell-Kelly
University of Warwick

Gordon Davies
Open University

John A. N. Lee
Virginia Tech

Michael R. Williams
University of Calgary

Prepublication Copy

1998 October

HISTORY IN THE COMPUTING CURRICULUM

Prepublication Copy

1998 October

CONTENTS

IFIP Joint Task Group (TC3/WG9.7)

HISTORY IN THE COMPUTING CURRICULUM

Prepublication Copy

1998 October

1. INTRODUCTION

History was always a part of the arts, mathematics, and the sciences. Many examples exist at the college or university level where courses such as history of music, history of mathematics, or cosmological history are a natural part of an education in these fields. When it comes to the history of the computing discipline, however, history seldom receives such status.

Since the 1960s, the ACM and the IEEE Computer Society have produced several curriculum recommendations. While these efforts are commendable, they lack or play down an essential component of computing education—history. Teachers often ignore history in computing when teaching computing courses; sometimes they mention it in passing. The typical case, however, is that they mention some facts and important milestones usually related to hardware, certain events, and the people responsible for these. The discussion then moves quickly to other topics. This is unfortunate, because we can learn much from history.

From a cultural standpoint, history broadens one's perspective on the field and lets students and scholars explore the inner thinking of people and the events they produced. From a practical standpoint, history enables individuals and enterprises to learn from the events of the past and to improve on experiences. Both views are necessary to create an informed computing professional. History should be considered as a part of human understanding and how the development of computing has affected the human environment. In more advanced settings, students should be able to do a critical analysis of significant moments in the history of computing.

2. GOALS AND OBJECTIVES

This report seeks to justify the establishment of a history component in the computing curriculum by providing a framework for a curriculum and resources. We intend to apply this objective to all undergraduate degree programs offered at any institution of higher learning. It is international in scope and we do not confine it to specific computing disciplines. This report also seeks to raise the awareness of history and how people can use it to improve the study and practice of the computing profession.

One of our goals in producing this curriculum is to establish an awareness of the history of computing in the computing field. This awareness leads to another goal, which is the importance of history as a tool for learning, both for the student and the practitioner. The awareness and importance of computing history lead to a third goal: The inclusion of history into the computing curriculum.

The goals outlined above are best manifested as a component of the formal education of students. This component can take one of these forms: an integration of history throughout the computing curriculum, a course designed for the computing specialists, or a course accessible to all non-computing specialists. Educators in the computing field need to learn the way history can make computing exciting. We need to give them the tools such as curriculum outlines, texts, resources, and sample curricula so they can easily include the study of history in the computing curriculum. The salient goal of this report is to make this happen.

3. OVERVIEW OF THIS REPORT

This report recommends guidelines on how history can be included as a part of computer education at the college or university level. The report suggests three ways of carrying this out:

❑ An integrated approach where history becomes a part of existing courses
❑ A single course for computing specialists who have completed at least one year of study in their specialty
❑ A single course with no prerequisite open to all students

These modes vary in the depth of presentation and the courses offered in the respective programs.

The approach taken by this report is to propose a knowledge base of suggested historical topics, including current events. Irrespective of the mode of implementation, the topics contained in the knowledge base suggest a possible course framework. Teachers can customize the knowledge base according to the needs of their individual programs.

For the integrated approach, teachers incorporate the historical topics into existing courses. For single-course implementations, the topics become the basis of the course. The course for non-computing specialists differs from the course for computing specialists by the depth in which each unit is presented or studied. While this proposal implies a complete (one-tenth year) course, reducing the length of a course by de-emphasizing certain topics. This possibility allows programs to include more easily history within their existing curriculum.

4. BACKGROUND

Before embarking on how history can add new dimensions to the study of computing, let us explore the development of the computing curriculum.

4.1 Overview of Curriculum Recommendations

The establishment of curriculum guidelines in computing have their roots from the early 1960s. In 1965 a committee of the ACM began to assemble topics for the emerging field of computing; three years later it published the first curriculum recommendations in computer science known as Curriculum'68 [1]. ACM updated these recommendations ten years later with the publication of Curriculum'78 [2]. In 1983 the IEEE Computer Society proposed curriculum recommendations for computer science and engineering programs [5]. In a joint effort in 1991, the ACM and the IEEE Computer Society developed curriculum guidelines for computing programs known as *Computing Curricula'91* [7]. ACM published in 1993 a four-volume set of curriculum guidelines for associate-degree programs covering the areas of computing sciences, information technology, computer support services, computing and engineering technology, and computing for other disciplines [3]. In 1997 several computing societies proposed curriculum guidelines for programs in information systems [6].

4.2 History Status

While the previously mentioned works are useful, little if anything is mentioned of the history of the subject. There is, however, reference to the social and ethical context of computing in some of these works, but no mention of how history can play an active role in the learning process. Teachers in the discipline have currently afforded a very low status to history of computing in the computing curriculum.

5. NEED FOR HISTORY CONTENT

5.1 The Student Perspective

For the student of computing, there is a need to understand how this field has developed and matured. How many computing students know about computing pioneers such as Grace Murray Hopper, John Atanasoff, Alan Turing, or Konrad Zuse? Students should be aware of the principal people, places, and events that shaped the profession. They should also be aware of the ideas, concepts, and ways of thinking that influenced its development. Students can learn from past experiences: the successes and the failures. They will become better professionals by learning more from the history of computing.

5.2 The Professional Perspective

Computing professionals can benefit from history. In an academic setting history adds new dimensions to courses, forces students to reflect on past events, and conceptualizes their academic studies of computing. In an industrial setting, practitioners can benefit from the study of history by being aware of past mistakes, misconceptions, and past successes. For an industry that has become very competitive, history can be viewed as a case study of events from the past and can be very rewarding.

6. CURRICULUM CONTENT

As already mentioned, a knowledge base for a history component in a computing curriculum consists of the significant historical events with a component reserved for current events. We now describe one such knowledge base.

6.1 Establishing a Knowledge Base

Many events identify the historical evolution of computing. Identifying which of these events are the most significant to a subject is often difficult. Nevertheless, as a start, we have identified some historical topics that we feel are important and that have had a significant influence on computing. Table 1 lists these topics. A listing such as described in Table 1 would form a knowledge base for an experience in computing history. Instructors are encouraged to formulate their own knowledge base for the experience they are presenting.

An extensive chronological listing of historical topics in computing may be found at the web site given in Appendix A. The topics of Table 1 can be combined with a selection from those in Appendix A. Any entry-level curriculum offering, whether an integrated or a single course approach, should include a knowledge base similar to that of Table 1. The emphasis on each event is left to the instructor.

Table 1

Suggested Knowledge Base of Historical Developments

Babbage's Engines
Census and Punched Card Machinery
Computability (Turing, Gödel)
Atanasoff-Berry Computer (ABC)
Zuse's Computing Machines
Use of Electronics (ENIAC, Colossus)
Stored Program and von Neumann
Invention of the Transistor
Invention of High-level Languages
Integrated Circuits
Operating Systems and System/360
VLSI Technology and the 4004 Chip
Parallel Computing
Personal Computers
The Internet
Human-Machine Interaction
Current Events

6.2 Methods of Presentation

An ideal situation would exist if all teachers of computing history possessed a sufficient and scholarly background in the subject. The reality is that there are relatively few computing historians. It will probably take some time before today's students will possess sufficient historical background to teach some day computing history.

To help establish a basis upon which first-time teachers of computing history can teach the subject, we offer several paradigms for presenting the material. A time-categorical overview is one method. Others are obviously possible. We provide these methods as a template for new teachers of the subject and to show how history can become a part of the computing curriculum.

6.2.1 The Period Method

The period method is one way to present the subject. It places people and events in time categories. Table 2 shows six suggested time categories and the rationale for each category. Teachers are encouraged to explore the use of different time periods that are better suited to their needs.

Table 2
Table of Historical Periods

Time category	Rationale
Before 1945	before the stored program concept
From 1945 to 1954	before high-level languages
From 1955 to 1970	before the invention of the chip
From 1971 to 1980	before the creation of the PC
From 1981 to the present	before the present (current events)

When these suggested time categories are coupled with the knowledge base of Table 1, it forms a template for the curriculum. The time categories and the important events within them provide a framework to help new teachers present this material.

Clearly, in the real world, topics will overlap in more than one category. For example, Fortran was developed in the early 1950s, but was not implemented commercially until the late 1950s. We use the arbitrary year 1955 as the transition between the two time categories. Therefore, the encapsulation from events in one category should not be treated in isolation of events that fall in other categories.

For most computing people, the time events or generations of hardware and software are already familiar to them. While it is possible to present the time-category method chronologically, there is no requirement to do it that way. The categories merely tend to encapsulate the events.

6.2.2 Other Methods

The method described in Section 6.2.1 is just one way to present computing history. Indeed, the chronological presentation of material may not be fostered by some of today's historians. Other exciting paradigms can be employed. Section 7.2 suggests three such methods to present a course for computing and non-computing specialists. Section 10.1 suggests an advanced presentation. Faculty are encouraged to explore different ways to present computing history and to use the method that best suits their philosophy, students, institution, and topic of discourse.

6.3 Depth of Knowledge

The depth of study of the curriculum is often based on the intended audience. The topics and activities shown in Section 6.2 can be presented at various levels of intensity. For a general computer history course for all students, one would expect a cursory discussion of topics with awareness of definitions and some conceptual understanding. When teaching computing specialists, whether by a single course or by an integrated approach, one would expect an ability to understand topics, to make comparisons, and to formulate applications. For intermediate and advanced teaching experiences, the expectation would extend beyond understanding and application and would gear more towards detailed accuracy and evaluation.

To help identify the level of audience and presentation, we include Table 3 that describes knowledge depth levels and their meanings. We base it in part on a taxonomy developed by the Software Engineering Institute [4].

Table 3
Depth Level and Meaning

Level	Meaning
0	No knowledge
1	Awareness through definition or recognition
2	Describe concepts and show understanding
3	Ability to differentiate and make comparisons
4	Apply knowledge to different settings
5	Formulate judgements and make evaluations

Application of this table is best served by using it to implement a history experience in the computing curriculum. The table can also serve to identify entrance prerequisites and exit competencies.

6.4 Clusters

To help educators, we have provided some examples of how they can develop a topic. We call these 'clusters' to show how they can include a topic in a curriculum. Clusters include an Identifier, a Subject, a Description, Topics, and Activities sections. A few comments under the heading Description describe the cluster. Under the heading Topics, we recommend a list of topics related to the cluster. Lastly, a history cluster contains a section called Activities, where we suggest projects or exercises. Appendix B illustrates examples of seven clusters.

7. IMPLEMENTATION OF A BASIC CURRICULUM

The topics from Table 1 form a knowledge base for including history in the computing curriculum. We can accomplish the inclusion through an integrated approach or by a single-course approach. One may offer single courses at a level for computing specialists or at a level of a general enrichment for others. An interdisciplinary approach with other departments that incorporates advice, guest lecturing, or team teaching is also possible and encouraged at all levels.

The purpose of an introductory experience to the history of computing is to give students a broad overview of the historical development of the subject. In addition, students will see the basics of historical analysis and writing. At the end of the experience, students should have an understanding of the key stages in the development of computing. They should also be able to study a relevant area of the history of computing in some depth. Samples of courses taught at some universities may be found in Appendix C.

7.1 The Integrated Approach

The integrated approach provides the opportunity for computing programs to include a history component. In this integrated approach, it is important that sufficient time is spent on history topics and for assessment. Modification of curriculum topics or the extension of these topics, even if it means adding more course time, can accomplish this. Since the audience is computing specialists, a prerequisite for this approach should be the successful completion of at least two courses (1/5 year) in computing.

We can integrate the topics of Table 1 into existing courses. For some programs, the integration may be obvious. In other programs, the integration may require more ingenuity. Whether it is a computer engineering or a data processing curriculum, making a meaningful inclusion into existing courses is possible. In each case, one should cover the topics of a knowledge base. Only the depth of exposure for each topic may vary from program to program. Implementation should ensure that approximately ten percent of each integrated course is devoted to history.

7.2 The Single-Course Approach

A single-course offering can easily achieve the inclusion of history of computing in a curriculum. Depending on the intended audience, such a course can be directed to computing specialists or to the general student body.

7.2.1 Course for Computing Specialists

A course in the history of computing intended for computing specialists should address the events in Table 1 plus a selection of the events referenced in Appendix A. We can accomplish the implementation of this course in a variety of ways. One way is to combine a theme of related events over time to make an interesting overview of the topics. For example, the study of the history of operating systems from the 1950s to date. Another approach could be a chronological progression of the subject. The prerequisite for this course is the successful completion of at least two courses in computing. The length of this course should be one-tenth year.

An example of this course is one that is offered at the University of Calgary. Instruction is given over a time period of 39 hours. The course assumes that students already know some elements of computing and have successfully completed the equivalent of two computing courses prior to taking the history course. A more detailed description of this course is found in Appendix D-1.

7.2.2 Course in Professionalism and Ethics of Computing

Another example of a course involving computing history is one that is offered at Virginia Tech. There, the course focuses on professionalism. Computing history and its effects on society are interwoven through the course giving the experience a dynamic perspective. A detailed description of the course is found in Appendix D-2.

7.2.3 Course for Students of Other Disciplines

A course for non-computing specialists differs from a course for specialists. Here, however, the depth or level of presentation and study is less than that prescribed in a course for specialists. The length of this course should be one-tenth year. There is no prerequisite for this course.

8. ASSESSMENT

As with all educational experiences, some form of assessment is necessary to measure and evaluate performance. Usually, we compare exit competencies and achievements with entrance prerequisites and with the modes of testing and evaluation. We recommend some guidelines as follows.

8.1 Entrance Prerequisites

Computing specialists exposed to an experience in the history of computing should have some knowledge of their field of study. As a result, whether we integrate the history experience in their studies or achieve it by a separate course, we must assume an entrance level. In these cases, the prerequisite depth of knowledge, as classified by the depth level described in Table 3, should be at least Level 1 and preferably Level 2. We also assume that each student has successfully completed two courses in computing.

For a general history of computing course designed for non-computing specialists, we assume no prerequisite. Here the entrance depth of knowledge is Level 0.

8.2 Testing and Evaluation

Written examinations, short essays, and a major essay would properly assess an experience in the history of computing. Typical examination questions include the following.

a) Outline the development of programming languages from the 1950s to date.
b) Explain the history and importance of the stored program concept.
c) Explain how the development of semiconductor electronics has affected the development of computers.

Typical essay assignments include a comparison of Babbage's calculating engines, the history of the BASIC programming language, the evolution of the object paradigm, the comparison of programming paradigms, biographical sketches, or the history of the personal computer. Students should expect to write an essay of five hundred to two thousand words on a topic of their choice.

8.3 Exit Competencies

We would expect students emerging from an experience on the history of computing to converse or to write on various topics of the field. They should relate concepts and topics with people, time, and places. For students who are computing specialists, the depth of knowledge competency should be at least Level 4 as described in Table 3. For others, the exit level should be at least Level 2.

9. RESOURCES

There are various means to obtain information today. Books, magazines, periodicals, videos, and the Internet are all accessible. Students and scholars should make ample use these resources. We illustrate some below.

9.1 Textbooks and General Works

We can use many books or texts for an introductory experience in the history of computing. Some texts that are suitable are by Augarten (1984), Campbell-Kelly and Aspray (1996), and by Williams (1997). Consult Appendix E for references to these and other materials.

9.2 Monographs and Articles

All issues of the *IEEE Annals of the History of Computing* can supplement these works. Appendix E suggests some possibilities in this area.

9.3 Electronic Information

Several Internet sites are useful in the study of computing history. One excellent site exists at Virginia Tech with URL <http://ei.cs.vt.edu/~history/>. For a list of other interesting web sites related to computer history, see Appendix F.

9.4 Videos, Simulators, and Other Resources

Videos are an excellent source for showing information to classes. Many videos on the history of computing show past events, show people and machines, and offer commentary on a variety of subjects. We show a list of some useful audio-visual aids that can complement discussions on computer history in Appendix G.

10. EXTENDED TOPICS

Developing scholars in the history of computing is advantageous for the computing profession. Teachers should give students avenues to increase their knowledge in this area and have mentors to guide them. To do this, further courses should be available to them. One can offer these courses as seminars or as tutorials. We now describe such courses.

10.1 Advanced Courses

Programs of study wishing to offer their students a more in-depth exposure to the history of computing, may wish to offer a course or seminar similar to the one now described. Final year students, even those with limited backgrounds in computing history, should have access to such a course. The course might have a title such as Intermediate Studies in the History of Computing. The prerequisite for this course is the equivalent of a mode already described in Section 8 of this report.

The purpose of an advanced course in the history of computing is to enable students to place the history of computing in a broad institutional, social, economic, and technological context. We would expect students to apply some methodologies and disciplines of social history, business and economic history, and the history of technology.

At the end of this course, students should be able to place the main stages in the development of computing in a broad historical context. They should also be able to write about a major theme in computing history based on one or more secondary sources.

At least one written examination, short essays, and a major essay would assess the course. Students may have written or oral examinations. Typical examination questions include:

a) In the 1950s there were two major contenders for an international scientific programming language: Fortran and Algol. Discuss reasons why Fortran became the standard language, despite being technically inferior to Algol.
b) Describe the role of the Moore School in establishing and distributing the stored program concept.
c) Discuss the claim that John V. Atanasoff is the "forgotten father of the computer."

Short and major essays are natural components of an advanced course. Some typical essay assignments would include: Alan Turing's contribution to computing, the development of rotational memory technologies, or the rise and fall of IBM from 1896 to 1990. Students should also produce a major essay of two- to four-thousand words. Teachers would require students to base this essay on a standard historical monograph, supplemented with some additional reading. Examples include Wexelblatt (1981), Stern (1981), and Hodges (1986). Other references are found in Appendix E.

Many resources are available for a course of this type. Besides a general text discussed in Section 9.1, teachers could require students to use one of the more serious general works such as Aspray (1990), Randell (1982), or Goldstine (1972). Teachers would also require students to use relevant articles from periodicals such as the *IEEE Annals of the History of Computing* or *Technology and Culture*. See Appendix D-3 for an example of an advanced course as given at the University of Warwick.

10.2 Projects

A project in the history of computing could also be available to advanced undergraduate and graduate students. The purpose of a project is to allow students to apply the methods of historical scholarship to the study of computing history and to allow for staff training and possible history department collaboration. It is also possible to include this advance material as an element of a History of Technology course in a history curriculum or with a Science Technology Studies program.

In this setting, students can undertake a significant self-directed historical inquiry. They can use historical sources effectively by displaying knowledge of some or all of the following: a) the distinction between primary, secondary, and tertiary sources, b) the effective use of periodical literature, c) the use of archives and corporate records, d) and the use of oral history. Students should be able to develop a dissertation of publishable quality as to research methodology and presentation (but not necessarily in depth or originality).

A dissertation of at least ten thousand words would assess a project. Examples of dissertation topics include the impact of technology on computer architecture, the history of the Internet, the evolution of objects, a technical and economic analysis of the software industry, or the controversy over the invention of the stored program concept.

11. CONCLUSIONS

The content of this report illustrates how the study of computing can include the study of computing history. Historical topics as listed in Table 1 form a suggested knowledge base. We show how one can include history in the curriculum by a systematic integration of historical events with existing courses or by the creation of self-contained history courses. We also make provision for a general education course for non- computing specialists. Included in this report are extensive information sources to achieve the goal to make the history of computing a viable component to any computer program of study.

12. FUTURE DEVELOPMENTS

The intent of this report is to provide teachers of the computing sciences a resource to help them infuse the subject of computing history in the curriculum. The authors recognize that this work is a first step and they encourage further development in this area such as current historical methodology and debates. The authors also believe that the educational community needs a sequel to this report to address the issue of “How to Teach Computing History.” It is their hope that this report will stimulate continued discussion, dialogue, and pedagogy that will enhance the inclusion of history in the computing curriculum.

ACKNOWLEDGMENTS

The Joint Task Group (TC3/WG9.7) of the International Federation for Information Processing (IFIP) prepared this report. The Joint Task Group wishes to thank Professor Wilfried Brauer of Technische Universität München and Professor Linton S. Thorn of Hofstra University for their comments and suggestions. It is also appreciative of the individuals and organizations that helped make this project possible.

REFERENCES

[1] ACM Curriculum Committee on Computer Science. "Curriculum '68 - Recommendations for Academic Programs in Computer Science." *Communications of the ACM 11* (3) (Mar. 1968): 151-197.

[2] ACM Curriculum Committee on Computer Science. "Curriculum '78 - Recommendations for the Undergraduate Program in Computer Science." *Communications of ACM 22* (3) (March 1979): 147-166.

[3] ACM Two-Year College Computing Curricula Task Force. *Computing Curricula Guidelines for Associate-Degree Programs*. 4 vols. New York: ACM Press, 1993.

[4] Ford, G. and Gibbs, N. A Master of Software Engineering Curriculum: Recommendations from the Software Engineering Institute. *IEEE Computer*, (Sept. 1989), 63.

[5] IEEE Computer Society Educational Activities Board / Model Program Committee. *The 1983 IEEE Computer Society Model Program in Computer Science and Engineering*. New York: IEEE Computer Society, 1983.

[6] Longenecker, Herbert E. Jr, David L. Feinstein, John T. Gorgone, Gordon B. Davis, J. Daniel Couger, co-chairs. *IS'97. Model Curriculum and Guidelines for Undergraduate Degree Programs in Information Systems*. ACM Press, 1997.

[7] Tucker, Allen B., et al. *Computing Curricula 1991 - Report of the ACM/IEEE-Computer Society Joint Curriculum Task Force*. New York: ACM Press, 1991. Reprinted in summary in *Communications of the ACM 34* (6) (June 1991): 68-84.

PUBLIC ACCESS

A copy of this report may be accessed by the Internet at the site <http://www.hofstra.edu/ComputingHistory/> or at the site <http://ei.cs.vt.edu/~history/>.

Appendix A

CHRONOLOGY OF HISTORICAL EVENTS IN COMPUTING

The chronology of historical events in computing may be accessed via <http://www.hofstra.edu/ComputingHistory/>. Additional information regarding this chronology is welcome. Individuals wishing to comment on the chronology should contact John Impagliazzo at <cscjzi@hofstra.edu>. Periodic monitoring of the site will ensure that historical information remains current and accurate.

Appendix B

SOME EXAMPLES OF CLUSTERS

CLUSTER 1

Subject: Software: Beginnings and Development

Description: Discussion of software landmarks from 1950 to 1975. The development of operating systems and programming languages are covered from the viewpoint of their effect on the growth and development of computing.

Topics:

- ❏ FORTRAN, the first high-level language and compiler
- ❏ LISP and its affect on artificial intelligence
- ❏ COBOL as a business language
- ❏ Early operating systems: Atlas, Exec II, CTSS, MULTICS
- ❏ The IBM 360 operating system
- ❏ ALGOL, its rise and fall
- ❏ The birth of BASIC, APL, PL/1, C
- ❏ SIMULA and object-oriented programming
- ❏ Other early software developments and Unix

Activities: Compare the differences between an early and a late operating system. Compare the differences between an early and a late programming language.

CLUSTER 2

Subject: Legal and Societal Issues in Computing

Description: Examples of legal issues and law suits and how their outcomes have affected or will affect the computing industry. Opposing positions on several cases are explored.

Topics:

- ❏ The Atari v. Nintendo case
- ❏ The Apple v. Microsoft case
- ❏ Computer theft and legal actions
- ❏ Legal versus ethics issues
- ❏ Social impact of computer ethics and law
- ❏ Human/computer interaction (HCI or CHI) issues
- ❏ Software/hardware changes due to HCI
- ❏ GUI interfaces

Activities: Research one or more past or present legal cases and summarize the opposing positions. Research the Code of Ethics of the ACM and the IEEE Computer Society and write an essay on their similarities and their differences.

CLUSTER 3

Subject: Mechanical Computing Devices before 1930

Description: Discussion of the development of mechanical calculating machines with the intent to show that easing the burden of computing was not just a goal of this century, but existed from very early times.

Topics:

- ❏ Simple calculating aids such as Napier's bones
- ❏ Mechanical machines such as those produced by Schickard, Pascal, Leibniz, etc
- ❏ The life and times of Charles Babbage:
    - ❏ His Difference Engine
    - ❏ His Analytical Engine
- ❏ Other mechanical computing systems such as Kelvin's tide predictors and Vannevar Bush's Differential Analyzer
- ❏ The work of Herman Hollerith and the development of punched card equipment
- ❏ Howard Aiken’s Calculations

Activities: Attempting to create and use a set of Napier's bones (or those of Genaille or Lucas for more advanced work). A comparison of the carry facilities on the machines of Schickard, Pascal, and Leibniz. Researching the uses made of Hollerith's equipment The careful study of the errors inherent in the mechanical analog computing instruments such as differential analyzers (for advanced students only)

CLUSTER 4

Subject: Automatic Computing Machinery from 1930 to 1945

Description: Discussion of the creation and use of the major calculating machines produced just prior to, and during, the second World War. Advanced students will want to have some appreciation of the development of relays and vacuum tube electronics during the early part of the 20th century.

Topics:
- ❏ The work of Konrad Zuse (Z1 through Z4)
- ❏ The work of Howard Aiken (Mark I and perhaps some mention of subsequent work)
- ❏ The work of George Stibitz (Model K, 1, and brief mention of 2,3,4 with a bit more information about the Model 5)
- ❏ The work of the code breakers - Enigma, the Bombe and Colossus
- ❏ The ENIAC - discussion of Eckert and Mauchly and others involved
- ❏ The ABC machine and Atanasoff

Activities: Use any of the various computer simulation tools that are now becoming available to try out an Enigma machine.
A comparison of the work of Zuse's Z4 and Aiken's Mark I and the common features. A comparison of the architecture of Aiken's Mark I and the ENIAC. An investigation of the topic of exterior ballistics and how this labor-intensive calculation was responsible for the ENIAC and other developments during this period.

CLUSTER 5

Subject: Hardware from 1945 to 1955

Description: Discussion of the creation of the tube-based stored program computer and how it developed in the first decade of its existence.

Topics:
- ❏ The Moore School's contribution to the development of the stored program concept - this might include the contributions of Eckert, Mauchly, von Neumann, and the impact of the design of the EDVAC and the Moore School Lectures
- ❏ The development of memory systems (delay lines, electrostatic systems, and the magnetic core)
- ❏ The computer development projects at the University of Manchester and Cambridge University (EDSAC)
- ❏ The NPL Pilot Ace machine and its non-von Neumann architecture
- ❏ The development of the "parallel" machine of von Neumann at the IAS (and its copies) and such machines as SWAC and UTEC
- ❏ The creation of the first computer corporations: Eckert-Mauchly (later UNIVAC), LEO, Ferranti, and IBM's eventual entry into the field
- ❏ The first generation of machines (UNIVAC, LEO, IBM 701, 702, Ferranti Mark I)
- ❏ The development of the Whirlwind and the consequences it had on both military and real-time computing
- ❏ The IBM 650 and the effect it had on bringing computing to the smaller institutions
- ❏ The transforming of the industry by transistor technology

Activities: Write a program for (or simply use a prewritten program) for one of the early machine simulators. A good simulator exists for EDSAC and several more are reportedly available. Compare early computer memory systems (Selectron, different delay mechanisms, other electrostatic systems). For advanced students, create a simple working model of a delay line memory system (one can be done using simple relays and a "slinkey" etc). Investigate the similarities between the IAS machine and the early IBM 701

CLUSTER 6

Subject: The Mainframe Era: Evolution of Architectures

Description: Discussion of the development of the mainframe computers and the shift from individual machine to the family of machines. Discussion of the mainframe evolution and its applications in industry.

Topics:
- ❏ The development of peripheral equipment (such as the IBM chain printers and random access disk systems) that permitted new areas of business to adopt computer techniques
- ❏ IBM and its computers
- ❏ The development of the IBM 360 series of machines
- ❏ The rise of the super machines - the product line of CDC and Seymour Cray
- ❏ The attempt by firms to emulate the product line of IBM with various families of machines
- ❏ The early development of timesharing
- ❏ The rise of DEC and their product line

Activities: Investigate just how compatible the machines from various ranges were. Attempt to produce comparisons between older machines and the ones that were available at the end of this era (eg the IBM 7090 and CDC 6600). Investigate and compare the operating systems of the mainframe machines (e.g. IBM 360 and CDC 6600). Investigate the storage capacities of different types of peripheral equipment available on machines of this era (tape drives, disk drives, cards) and compare them to modern storage facilities. Investigate some of the more unusual peripheral devices (the NCR CRAM file, the photo-store used at Lawrence Livermore Labs).

CLUSTER 7

Subject: Early Computing Concepts, Algorithms, and Models

Description: Discussion of the development of the early development of the theories, concepts, and notions surrounding computing. The focus is on people and their concepts.

Topics:
- ❏ Turing: computable functions and machine models
- ❏ Post: machines as logic oriented and symbolic manipulation

paradigms

- Gödel, Herbrand, Kleene: notions of recursive functions defined by equation systems or operators
- Church, Rosser: lambda calculus and computable functions
- Shannon, Zuse: use of logical and algebraic foundations to model switching circuits
- McCulloch, Pitts: models for neurons and neural networks
- Markov: notion of an algorithm as a deterministic string-rewriting system (development of SNOBOL)
- Von Neumann: cellular automata and the notion of massively parallel computers
- Holland: genetic algorithms
- Petri: invention of Petri nets as a basis of the theory of concurrency
- Chomsky: formal languages and grammars

Activities: Investigate one or more individuals mentioned in the topics of this cluster. Write a short essay that elaborates on their early notions, ideas, or inventions and show how those concepts have affected today's computing.

Appendix C

SOME SITES FOR COMPUTING HISTORY COURSES

Below is a sample of computing history courses taught at different universities. A more complete list may be found at the web site <http://www.dcs.warwick.ac.uk/~mck/HoC_Courses.html>, which will be updated periodically by Martin Campbell-Kelly. Feel free to contact him at <mck@dcs.warwick.ac.uk> for comments.

Institution:	University of Warwick
Lecturers:	Martin Campbell-Kelly & Steve Russ
Course Title:	CS330 History of Computing
URL:	<http://www.dcs.warwick.ac.uk/~mck/HoC_Courses.html>

Institution:	University of Calgary
Lecturer:	Michael R. Williams
Course Title:	CPSC 509 History of Computation
URL:	<http://www.cpsc.ucalgary.ca/Courses/defaults/509/>

Institution:	University of Stanford
Lecturer:	Paul N. Edwards
Course Title:	STS161 History of Computers
URL:	<http://www-leland.Stanford.edu:80/group/STS/161.shtml>

Institution:	University of Swansea
Lecturer:	John V. Tucker
Course Title:	CS309 History of Computation
URL:	<http://www.swan.ac.uk/compsci/HandbookFolder/courses/level3/CS_309.html>

Institution:	University of Minnesota
Lecturer:	Arthur L. Norberg
Course Title:	CSci5090 History of Computing
URL:	<http://www-users.itlabs.umn.edu/~tran0177/csug/courseDescriptions/TOcsci5xxx.html>

Institution:	University of Michigan
Lecturer:	Eric Klavin
Course Title:	EECS 490 History of Computing - Michigan
URL:	<http://www-personal.engin.umich.edu/~klavins/hoc/info.html>

Institution:	Cornell University
Lecturer:	Michael A. Dennis
Course Title:	S&TS 355 Computers: From Babbage to Gates 1.0
URL:	<http://instruct1.cit.cornell.edu/courses/sts355/WebPageSTS355.htm>

Institution:	Princeton University
Lecturer:	Michael S. Mahoney
Course Title:	History 398 Technologies and Their Societies: Historical Perspectives
URL:	<http://gardens.princeton.edu/~mike/h398/398syl.html>

Appendix D-1

COURSE SYLLABUS SAMPLE

UNIVERSITY OF CALGARY

Title: The History of Computation

Duration: 39 hours of lectures, three examinations, (1/10 year of study)

Description: This course is designed to give the student an appreciation of the long history of human need to ease computation. It is not the history of "computers" but looks at all aspects of computation from early systems of numeration, through development of mechanical means of calculation, to the development of the microprocessor chip. While the course is oriented around the technology of computation, it does attempt to show the social, political, and economic background that drove the developing technology.

Text:
1. *A History of Computing Technology* by Michael R. Williams, IEEE Computer Society, 1997.
2. Additional material is provided in the form of articles in the journals, mainly from *Annals of the History of Computing*, web pages, and class handouts.

Syllabus:

Systems of Numeration in ancient and modern societies
Written Number Systems
- The Additive Number System, The Positional System
- The Egyptians, The Greeks, The European Number System, The Far East

Other Forms of Notation
- Knotted Cords for Record Keeping, Tally Sticks

Early Aids to Calculation
Finger Reckoning
The Abacus
The Quadrant
The Proportional Compass
The Sector
Napier's "Bones"
- Gaspard Schott and Athanasius Kircher
- Genaille--Lucas Rulers

Logarithms - The Slide Rule
Mechanical Calculating Machines
- Wilhelm Schickard, Blaise Pascal, Gottfried Wilhelm Leibniz, Samuel
- Morland, Ren Grillet, The Thomas Arithmometer, The Baldwin-Odhner Machines, Key-Driven Machines

Charles Babbage (1791 - 1871) and His Machines
- The Method of Differences
- Babbage's Difference Engine
- The Scheutz Difference Engine
- Babbage's Analytical Engine
- Percy Ludgate

The Analog Methods and Machines
- The Astrolabe
- The Antikythera Device
- Tide Predictors
- Differential Analyzers

The Large Mechanical Calculators
- The Zuse Machines: Z1, Z2, Z3, Z4
- The Bell Relay Computers:
- The Harvard Machines of Howard Aiken

The IBM Calculators
- The Punched Card Systems
- The Large IBM Calculators
- The Selective Sequence Electronic Calculator (SSEC)

The Electronic Revolution
- John Atanasoff, Clifford Berry, & the ABC
- The ENIAC: The Place and the Problem, The People, The Machine
- The Colossus Machines
  - The Enigma, Alan Turing, The Robinsons, The Colossus

The First Stored Program Electronic Computers
- The Genesis of the Ideas

Computer Memory Systems
- Thermal Memories, Mechanical Memories, Delay Line Systems, Electrostatic Storage Mechanisms, Rotating Magnetic Memories, Static Magnetic Memories

The British Scene
- The Manchester Machine
- The Cambridge Machine - EDSAC
- The NPL Pilot Ace

The American Scene
- The American Background
- The Electronic Discrete Variable Arithmetic Computer (EDVAC)
- The Institute for Advanced Study Machine (IAS)
- The Eckert/Mauchly Machines, BINAC and UNIVAC
- The SEAC and SWAC Machines
- Project Whirlwind
- Later Developments
- The Early Machines of IBM: The NORC, The 700-7000 Series Machines
- Early Super Computers: The Stretch, The LARC, The Ferranti Atlas

The IBM/360 Series of Machines
The Development of the Microprocessor

Appendix D-2

COURSE SYLLABUS SAMPLE

VIRGINIA TECH

(Shows how history can be embedded in a broadening course.)

Title: Professionalism in Computing

Description: This course studies the social impact, implications and effects of computers on society, and the responsibilities of computer professionals in directing the emerging technology. Includes examinations of reliable, risk-free technologies, systems which provide user friendly processes. Specific topics include an overview of the history of computing, computer applications and their impact, the computing profession, and the legal and ethical responsibilities of professionals. The course is taught in two parts - Part I: Professionalism and Part II: Careers. This portion of the syllabus only refers to Part I.

Objectives: Having successfully completed this course, students will be able:

1. To review and analyze the effects of the insertion of computer technology into society, and to anticipate the impact of that technology on people, companies and the community;
2. To select from the many algorithms for the implementation of computer applications those that will not only satisfy the needs of economy but also those that will have higher factors of safety, greater sensitivity to user needs, and increased reliability;
3. To use the underlying concepts of both technology and ethical behavior to realize their own potential impact on the community that they serve, and to make rational decisions regarding their responsibilities to the community.

Textbooks:

1. Bowyer, Kevin W. ETHICS and Computing: Living Responsibly in a Computerized World, IEEE Computer Society Press, Los Alamitos CA, 1996, xvi, 449, paperback.
2. Williams, Michael. A History of Computing Technology, IEEE Computer Society Press, 2nd Ed., 1997.

Prerequisites: A depth of knowledge of computers sufficient to understand the implications and impact of applying a computer to everyday situations.

Assignments: There will be four assignments to be handed-in for grading in this course, together with participation in making individual presentations and in class debates. This is a writing intensive course required for all CS majors.

Examinations: This course is heavily loaded with out-of-class activities and writing assignments. Thus there is only one examination - the final examination.

Grading: The final grade will be based on these factors: 1. The written assignments (30%), 2. The debate (15% for presentation, 5% for the report), 3. Class participation (30%), 4. The final examination (20%)

Online Resources: The digital library of materials pertinent to this course, including course notes and slides are available on the web at: http://ei.cs.vt.edu/~cs3604/. You should check this site bi-weekly for updates and information.

Appendix D-3

COURSE SYLLABUS SAMPLE

UNIVERSITY OF WARWICK

Title: History of Computing

Text:

Aspray and Campbell-Kelly; *Computer: A History of the Information Machine*, Basic Books, 1996.
A dozen historical articles for study in depth of a specific topic.

Duration: The credit for this course is 15 CATS points taken in the final year of study.

Aims: To give the student (a) a background to the technical and commercial development of computing and information technology, (b) a training in historical methodology and scholarship.

Content: The course will be taught be means of lectures, based around selected readings in the history of computing. Lectures will be supplemented with visual aids and occasional seminars by outside speakers. Assessed work will consist of an introductory assignment, plus a major essay assignment or programming/ mathematics project.

Topics:
- Early aids to calculation: development of mathematical tables and other aids to calculation
- Calculating engines: Babbage's difference and analytical engines; the difference engines of Scheutz and others
- Logic and languages: the development in mathematical logic and formal languages, from Leibniz to Gödel, which provided a framework for the notion of computability and the definition of programming languages
- Algorithms and computation: an account of the remarkable confluence of ideas in the 1930s which culminated in the Church-Turing thesis and in the work and machines of Alan Turing.
- Punched-card machinery: Hollerith's census machine; development of punched-card machinery; commercial and mathematical exploitation
- The early IT environment: office machinery and the office-machine industry; communication systems
- Relay calculators: the Harvard Mark I and other machines.
- Electronic calculators: the ENIAC, cryptanalytic machinery, electronic punched-card machinery
- Early stored program computers: EDSAC and other machines.
- Development of computing and applications: selected developments in hardware, software, theory and applications, from 1950 onwards
- Third generation computers: the rationale for the IBM System/360; competitive responses
- Development of the computer industry: comparative development in the United States and elsewhere; development and rationalization of the British computer industry.
- Personal Computing: the technical, economic, and cultural development of the personal computer.

Appendix E

JOURNALS, TEXTBOOKS, GENERAL WORKS, AND MONOGRAPHS ON COMPUTING HISTORY

Journals

IEEE Annals of the History of Computing; IEEE Computer Society, Los Alamitos, California.

Technology and Culture; Society for the History of Technology, Dearborn, Michigan.

Textbooks and General Works (Not all books are in print, but are available in libraries.)

Aspray, William (Ed.); *Computing Before Computers*, Iowa State University Press, 1990.

Augarten, Stan; *Bit by Bit: An Illustrated History of Computers*, Ticknor & Fields, 1984.

Bauer, Friedrich L., Brauer, Wilfried, and Schwichtenberg, Helmut, eds; Distributed Action Systems. In: *Logic and Algebra Specification*, Springer-Verlag, Berlin, pp.1-30, 1993.

Brauer, Wilfried; The New Paradigm of Informatics. In: *New Results and New Trends in Computer Science*, Lecture Notes in Computer Science, Vol 555, pp. 15-24, Springer-Verlag, 1991.

Campbell-Kelly, Martin and Aspray, William; *Computer: A History of the Information Machine*, Basic Books, 1996.

Freiberger, Paul; *Fire in the Valley: The Making of the Personal Computer*, Berkeley, 1984.

Goldstine, Herman; *The Computer from Pascal to von Neumann*, Princeton University Press, 1972.

Kidder, Tracy; *The Soul of a New Machine*, Boston 1981.

Lee, John A.N.; *Computer Pioneers*, IEEE Computer Society Press, 1995.

Levy, Steven; *Hackers: Heroes at the Computer Revolution*, Garden City, 1984.

Randell, Brian (Ed.); *The Origins of Digital Computers*, Springer-Verlag, 1982.

Sobel, Robert; *IBM: Colossus in Transition*, Boston, 1981.

Williams, Michael R.; *A History of Computing Technology*, IEEE Computer Society Press, 1997.

Specialist Monographs

Austrian, Geoffrey D.; *Herman Hollerith: Forgotten Giant of Information Processing*, Columbia University Press, 1982.

Bergin, T.J. and Gibson, R.G.; *History of Programming Languages*, ACM Press, New York, 1996.

Hodges, A.; *Alan Turing: The Enigma*, Simon and Schuster, New York 1983.

Pugh, Emerson W.; *Building IBM: Shaping an Industry and its Technology*, MIT Press, 1995.

Pugh, Emerson W., Lyle R. Johnson, and John H. Palmer; *IBM's 360 and Early 370 Systems*, MIT Press, 1991.

Pugh, Emerson W.; *Memories That Shaped an Industry*, MIT Press, 1984.

Stern, Nancy; *From ENIAC to UNIVAC: An Appraisal of the Eckert-Mauchly Computers*, Digital Press, 1981.

Wexelblatt, R. (ed.); *History of Programming Languages*, Academic Press, 1981.

Appendix F

WEB SITES USEFUL TO COMPUTING HISTORY

Some web sites useful in the computing curriculum

<http://www.computer.org/pubs/annals/annals.htm >
Home page for the IEEE Annals of the History of Computing.

<http://ei.cs.vt.edu/~history/index.html >
The starting point for a lot of stuff.

<http://www.cbi.umn.edu >
Charles Babbage Institute, Center for the History of Computing. University of Minnesota.

<http://www.cs.man.ac.uk/CCS >
Computer Conservation Society. British Computer Society and the Science Museum of London.

<http://www.tcm.org/ >
The Computer Museum, Boston, Massachusetts

<http://www.comlab.ox.ac.uk/archive/other/museums/computing.html >
The Virtual Museum of Computing, Oxford University

<http://www-groups.dcs.st-and.ac.uk/~history/ >
University of St. Andrew’s MacTutor History of Mathematics archives.

<http://www.compustory.com >
The American Computer Museum in Bozeman, Montana.

Some web sites that may be of interest

<http://www.mhs.ox.ac.uk >
Museum of the History of Science, Oxford, England

<http://www.webcom.com/calc/ >
Calculating Machines. The History of Calculating.

<http://www.geocities.com/Athens/Delphi/9810/lostmuseum.html>
Lost Museum. Go to the "Computer Exhibit" to access a variety of interesting historical information.

<http://www.dcs.warwick.ac.uk/~edsac/ >
EDSAC simulator home page

<http://info.isoc.org/guest/zakon/Internet/History/HIT.html >
Hobbes' Internet Timeline.

<http://www.geocities.com/SiliconValley/2260 >
NetHistory.

<http://www.best.com/~thvv/7094.html >
The IBM 7094 and CTSS.

<http://www.ics.uci.edu/~eppstein/numth/egypt/ >
Discusses Egyptian fractions and the systems used to generate them.

<http://www.mailcom.com/besm6 >
BESM-6 Nostalgia Page.

Appendix G

AUDIO-VISUAL MATERIALS

Video Tapes and Slides (Not all materials may be available for purchase, but are accessible in libraries.)

"Computer Pioneers and Pioneer Computers", video tape of excerpts from the talks given by various pioneers at The Computer Museum in Boston, Volume 1 "Dawn of Electronic Computing: 1935-1945". Volume Two "The First Computers: 1946-1950", 1996.

"Wood, Brass and Baboon Bones", BBC Open University video tape which looks at calculating and recording devices from prehistoric times (the "Inshango" bone) to the time of Babbage. Available from the Open University and from the Science Museum in London.

"The Machine that Changed the World", (Five-parts), the WGBH Collection, films for the Humanities & Sciences, Princeton, 1992.

“Minerva’s Machine - Women and Computing”, Karen A. Frenkel, PBS and ACM, New York, 1995.

“EDSAC”, Cambridge University Press, 1976/1951.

"The History of Silicon Valley", PBS video tape.

"Alan Turing: Breaking the Code", PBS video tape.

“John von Neumann”, MAA, 1966.

"Nova: Code breakers", PBS video tape.

Note: Slide sets are available from the Science Museum in London and The Computer Museum in Boston.

IFIP JOINT TASK GROUP (TC3/WG9.7)

HISTORY IN THE COMPUTING CURRICULUM

John Impagliazzo
(Task Group Chair)
Department of Computer Science
103 Hofstra University
Hempstead, New York 11549-1030
United States
Phone: +1 516-463-6774
Fax: +1 516-463-5790
Email: cscjzi@hofstra.edu

Martin Campbell-Kelly
Department of Computer Science
University of Warwick
Coventry CV4 7AL England
United Kingdom
Phone: +44 203 523193
Fax: +44 203 525714
Email: mck@dcs.warwick.ac.uk

Gordon Davies
Computing Department
Faculty of Mathematics and Computing
The Open University
Milton Keynes MK7 6AA England
United Kingdom
Phone: +44 1908 652680
Fax: +44 1908 652140
Email: g.davies@open.ac.uk

John A.N. Lee
Department of Computer Science
Virginia Tech
Blacksburg, Virginia 24061-0106
United States
Phone: +1 540-231-5780
Fax: +1 540-231-6075
Email: janlee@cs.vt.edu

Michael R. Williams
Department of Computer Science
University of Calgary
2500 University Drive, NW
Calgary, Alberta T2N 1N4
Canada
Phone: +807 403-220-6781
Fax: +807 403-284-4707
Email: williams@cpsc.ucalgary.ca