# Undergraduate Computer Science and Engineering Curriculum in India

Vaidyeswaran Rajaraman

*Abstract*—**The undergraduate computer science and engineering degree in India is a professional engineering degree and follows the general structure of other engineering degree programs. It aims to provide a good breadth in basic engineering and 50% of the curriculum in common with all engineering disciplines. The curriculum has a strong electrical engineering bias. At present there is no formal accreditation of engineering programs in India and each university is expected to maintain their own standards. This paper describes the general structure of engineering education in India and the philosophy used in evolving an undergraduate curriculum in computer science and engineering.**

## I. Introduction

IN India, the following applications of computers are considered very important:

- Use of micro and minicomputers in process control to improve productivity and save energy consumption.
- Computer aided design in engineering industry.
- Decision support systems in industry and government. In particular, the development of systems which would aid in production planning, inventory control and maintenance management.
- Application of computers in improving public transport systems such as railways, airlines, public utilities, and telecommunication systems.
- Developing a computer industry which would design and manufacture computers, develop software, and maintain these computers.
- Developing an applications software industry.
- Self reliance in maintaining hardware and software of imported computers.

From the list above, it is clear that human resources with a multidisciplinary background to develop sophisticated applications software and for providing the leadership to develop a computing industry are needed. A variety of courses to meet these goals are offered in India. In what follows, I will primarily discuss the development of a curriculum for an undergraduate program in computer science and engineering.

## II. A Bird's Eye View of the Engineering Education Scene

The Computer Science and Engineering undergraduate degree in India it is on par with any other engineering degree and follows the same general structure. Engineering education in India has the following features:

- Students join an engineering college after 12 years of primary and secondary school education.
- In order to join engineering the student is required to have studied physics, chemistry, and mathematics in the last four years of school and have spent approximately 60% of their time studying these subjects.
- Engineering education is heavily subsidized by the state and the tuition fee in most colleges is low. The education budget of most states is low. Thus, the quality of the available infrastructure such as computing laboratory, library, and other educational material is poor.
- Typically the computer science and engineering department of engineering colleges would have around 20 IBM compatible PC's (mixture of XT's and AT's) and a time shared, minicomputer running Unix. They normally do not have graphics workstations such as SUNSPARC or IRIS.
- There are approximately 130 engineering colleges in the country. Approximately 100 of them offer a bachelor's degree in computer science and engineering. About 3000 students graduate each year with a bachelor's degree in this discipline.
- There are five Indian Institutes of Technology (IIT) which are very well equipped and their educational pattern is science and research oriented. They admit 1500 undergraduate students each year out of around 100 000 students who compete for these places by taking a difficult entrance examination. The number of students admitted to computer science and engineering is about 150.
- The duration of the undergraduate engineering course is four years.
- Groups of engineering colleges (average four to five) are affiliated with a university. The university constitutes a committee consisting of teachers from the university and other universities and professionals who formulate and approve the curriculum.
- The IIT's and a few other well established institutions formulate their own curriculum with faculty committees and through informal consultation with professionals.
- Many universities have courses extending over an academic year (38 weeks) with an examination at the end of each year. IIT's and a number of universities, however, have a semester pattern. A semester typically extends over 16 weeks. An academic year has two semesters. Courses are not normally taught in summer months.

Manuscript received June 1992.

The author is with the Supercomputer Education and Research Centre, Indian Institute of Science, Bangalore 560012, India.

IEEE Log Number 9202865.

- At present, there is no formal accreditation of engineering colleges. Universities are expected to maintain the standards of their affiliated colleges.
- Recently, many new engineering colleges have been started which are not funded by state governments. Many of these do not have adequate facilities. Quality control through formal accreditation is now being initiated. An All India Council for Technical Education has been formed by the Government of India and has been given the responsibility for accreditation of both state and private engineering colleges. Accreditation will be based on curriculum, quality of faculty, students, and infrastructure.
- The Indian Society for Technical Education—whose members are engineering college teachers—plays an important role in formulating model curricula in various engineering disciplines.
- All engineering students take a common set of compulsory courses in english, physics, mathematics, chemistry, engineering, humanities, and social sciences.
- Compulsory engineering courses are designed to give breadth in engineering and to enable students to understand sound engineering principles in design.
- The common core courses take up 50% of the available time.
- All engineering students have a design project during the last year in which the students design and build a small hardware or software system, document its design and make an oral presentation to a board of examiners.
- Engineering students wishing to pursue post-graduate study take an examination known as Graduate Aptitude Test in Engineering (GATE). This has one common subject for all engineers and an advanced subject in the area of specialization. Undergraduate curricula are designed to enable students to take this test.

## III. Philosophy of Undergraduate Education

The main objective in the design of an undergraduate curriculum is to provide breadth of understanding of basic science and engineering design principles. An engineer's main concern is to synthesize products systematically using sound design principles to meet specifications of performance and cost. This principle is valid for both hardware and software products. An undergraduate's education must provide solid grounding for life long learning. Thus, the core must consist of topics which provide the foundation. Importance of documentation, design for maintainability, quality control, reliability, division of labor, etc., which form the cornerstone of good engineering education are also important in software education. An actual design project which aims to synthesize topics learned in different courses and which leads to a prototype is a great motivator for the students and has immense educational value.

Basic education principles in engineering education are not totally dependent on the country where it is taught. The social concerns and local constraints do, however, affect the detailed choice of courses in a curriculum [1].

The basic sciences are necessary for their own sake and to provide grounding for engineering science courses such as mechanics of solids, thermodynamics, and electronics. As the entering students already have a reasonably good grounding in physics, chemistry, and mathematics, science courses in engineering curricula in India are not many.

It is the general consensus in India that the distribution of subject area as a percentage of total time spent in a four year engineering course should be:

| | |
|---|---|
| Basic Sciences and Mathematics | 20% |
| Humanities and Social Sciences (including English) | 10% |
| General Engineering | 20% |
| Engineering Specialization | 50% |

Most curricula are designed using these general guidelines.

In the engineering specialization approximately 35% is devoted to compulsory courses which form the basic "non-changing" part of the subject. The rest are devoted to electives and project work. Computer science and engineering is a fast changing field. Thus, it is a challenge to pick the compulsory courses to meet the twin objectives of providing breadth in the subject and a good foundation to the subjects taught.

## IV. Undergraduate Curriculum Design

The first undergraduate computer science and engineering course at an IIT was started in 1978 at Kanpur. The curriculum designed at IIT, Kanpur, had a core in common with all other engineering courses. It had some electrical engineering courses such as digital electronics, signals and systems, and communication systems. The computer science core consisted of data structures and algorithms, automata theory, programming languages, operating systems, computer architecture, etc. Based on the recommendations of a manpower committee [2] appointed by the Government of India, many other engineering colleges also started an undergraduate course in computer science and engineering. Most institutions formulated curriculum similar to that of IIT, Kanpur. By 1984 many universities had started such undergraduate courses and it was felt that a model curriculum should be evolved for the guidance of engineering colleges. The Ministry of Human Resource Development, Government of India (which funds a large number of technical institutions including the IIT's) requested the Indian Society for Technical Education to formulate a curriculum. An expert group was formed with representatives of academic professional societies (Computer Society of India and Institution of Electronics and Telecommunication Engineers), and industries. This group conducted meetings in many cities of India, evolved a curriculum and elicited comments on it. A final model curriculum was accepted and published in 1987 [3] which has guided the evolution of most curricula in the past four years.

In the suggested curriculum, the courses in the first three semesters are common to all engineering disciplines. The rest has mostly computer science and engineering subjects and some courses in mathematics, electrical engineering, and humanities and social sciences.

The distribution of credit hours in the four year curriculum recommended by the group is:

| | |
|---|---|
| Physics and Chemistry | 15 |
| English, Humanities and Social Sciences | 19 |
| Mathematics | 20 |
| General Engineering | 30 |
| Electrical Engineering | 19 |
| Computer Science and Engineering (including project) | 98 |
| Total | 201 hours |

The details are given in Appendix I.

In evolving the core courses, it was agreed that courses should cover the essential subjects of computer science and engineering, namely the following:

- algorithms, data structure, and problem solving;
- automata, languages, and computation;
- programming languages and their translators;
- logic design of computers;
- computer architecture;
- operating systems;
- information systems and data bases.

Besides this, electives are provided in areas such as artificial intelligence, distributed computing, computer graphics and robotics to let a student take topics that would be useful to carry out the final year bachelor's degree project. The structure of the curriculum is closer to the computer engineering curricula in the United States than to the computer science curricula. When compared to the ACM, IEEE Computer Society suggestion [4] of nine subareas of computing discipline seven are covered in the core of compulsory courses in the curriculum. Two areas, namely, artificial intelligence and robotics and human-computer communication are not in the compulsory core but are among the suggested electives. An interesting point to note is the necessity for formal weekly laboratories in computing subjects have been recognized in this curriculum as has been done in [4] and is incorporated in working out the credit hours of courses. The nucleus of the compulsory courses to be taken by students of computer science and engineering and the semester in which they are taken are given as follows:

| *Course* | *Semester* |
|---|---|
| THEORY | |
| 1. Discrete Mathematical Structures | IV |
| 2. Probability, Statistics and Queueing Models | IV |
| 3. Automata, Languages, and Computation | V |
| 4. Signals and Systems | V |
| SOFTWARE | |
| 1. Problem Solving and Computing | III |
| 2. Data Structures and Programming Methodology | IV |
| 3. Systems Programming | V |
| 4. Principles of Programming Languages | VI |
| 5. Operating Systems | VI |
| 6. Algorithm Analysis and Design | VII |
| HARDWARE | |
| 1. Logic Design | III |
| 2. Electronics and Instrumentation | IV |
| 3. Computer Organization and Architecture | IV |
| 4. Digital Electronics | V |
| 5. Microprocessor Systems and Applications | VI |
| APPLICATIONS | |
| 1. Data Processing and File Structure | V |
| 2. Data Communication | VI |
| 3. Information Systems Design | VII |

Besides these courses two compulsory courses, one on "Computer and Society" and the other on "Technical Writing and Communication" are suggested for semesters VII and VIII, respectively, in the model curriculum. The detailed curriculum for each subject is given in [3].

## V. Other Programs in Computing

Besides the Bachelor's degree in computer science and engineering there are other courses offered by universities at the post-graduate level.

- A Master of Computer Applications (MCA) course is a three year course intended for graduates in science or commerce. The primary emphasis of this course is on information systems development for management [5].
- A Master of Engineering course in computer science and engineering is a three semester course and is open to students who pass a Graduate Aptitude Test in Engineering (GATE) which is similar to the GRE in the U.S. context. The number of students admitted to this course by the IIT's and other universities is around 300 a year. In many universities, this course is open not only to Bachelor's degree holders in computer science and engineering, but also to other engineering graduates.
- The total number of computer science and engineering students graduating each year is around 3000. Thus there is a serious human resource shortage in this area which has adversely affected the growth of industry. Recently, the Department of Electronics of the Government of India, industries, professional societies (The Computer Society of India and the Institution of Electronics and Telecommunication Engineers) and the Ministry of Human Resource Development, have formed a National Advisory Committee to accredit private institutions to conduct computer courses. Curricula for various levels of education beginning with basic computing, systems analysis and design and bachelor's and master's courses, are being formulated. National tests will be conducted by the professional societies which can be taken by students to get an equivalent of a Bachelor's or Master's degree. The details of curricula and tests are being worked out.

## VI. Conclusion

The major difference between the undergraduate computer science and engineering curriculum in India and in the West is an electrical engineering bias in our curriculum. Sound engineering grounding we feel is important in designing software, particularly, software embedded in systems.

The major problem in India is the dearth of qualified teachers to teach computer science and engineering. The job opportunities in industry are very attractive and it is difficult to attract qualified persons to teach. To alleviate the shortage of teachers, the government of India started a teacher training program. Students graduating with a Bachelor's degree in engineering who wanted to become teachers of computer science were given scholarships to obtain a Master's degree in computer science and engineering and offered teaching positions. The number of teachers trained was very small (around 30) and was not sufficient to meet the demand. The dearth of teachers is partly alleviated by hiring part-time teachers from industry and by recruiting fresh graduates as teaching assistants. This is a stop-gap measure and it is expected that with more schools offering Master's degrees, the teacher shortage will be gradually solved. The quality of students joining computer science course is very good as they are admitted after a tough competitive examination. As many good text books are available and computing resources are also accessible, many students are able to perform very well by self-study and discussions with fellow students. A good detailed curriculum document with suggestions on self-study material and text books is thus crucial.

## Appendix I

### Model Curriculum Details for a Four Year Degree Course in Computer Science and Engineering

| *SEMESTER I* | *Lectures* | *Tutorials* | *Practicals* |
|---|---|---|---|
| 1) Language I (English) | 2 | 1 | 0 |
| 2) Mathematics I (Calculus and Analytic Geometry) | 3 | 2 | 0 |
| 3) Physics I | 3 | 1 | 2 |
| 4) Chemistry | 3 | 1 | 2 |
| 5) Workshop Practice I | 1 | 0 | 3 |
| 6) Engineering Graphics I | 0 | 0 | 3 |
|  | 12 | 5 | 10 |

| *SEMESTER II* |  |  |  |
|---|---|---|---|
| 1) Language II (English) | 2 | 1 | 0 |
| 2) Mathematics II (Differential Equations) | 3 | 2 | 0 |
| 3) Physics II | 3 | 1 | 2 |
| 4) Workshop Practice II | 1 | 0 | 3 |
| 5) Engineering Graphics II | 0 | 0 | 3 |
| 6) Mechanics I (Statics and Rigid Body Dynamics) | 3 | 1 | 1 1/2 |
| 7) Thermodynamics and Heat Transfer | 3 | 2 | 0 |
|  | 15 | 7 | 9 1/2 |

| *SEMESTER III* |  |  |  |
|---|---|---|---|
| 1) Mathematics III (Vector Calculus) | 3 | 2 | 0 |
| 2) Problem Solving and Computer Programming | 3 | 1 | 2 |
| 3) Mechanics II (Mechanics of Solids) | 3 | 1 | 1 1/2 |
| 4) Electrical Science I (Basic Electrical Engineering) | 3 | 1 | 2 |
| 5) Material Science and Engineering Materials | 3 | 1 | 2 |
| 6) Logic Design | 3 | 1 | 0 |
|  | 18 | 7 | 7 1/2 |

| *SEMESTER IV* |  |  |  |
|---|---|---|---|
| 1) Electrical Science II (Electronics and Instrumentation) | 3 | 1 | 3 |
| 2) Introduction to Engineering System Design | 2 | 1 | 0 |
| 3) Discrete Mathematical Structures | 3 | 1 | 0 |
| 4) Probability, Statistics and Queueing Models | 3 | 1 | 0 |
| 5) Data Structures and Programming Methodology | 3 | 1 | 3 |
| 6) Computer Organization and Architecture | 3 | 1 | 3 |
|  | 17 | 6 | 9 |

| | | | |
|---|---|---|---|
| *SEMESTER V* | | | |
| 1) Humanities and Social Sciences (Elective) | 2 | 1 | 0 |
| 2) Signals and Systems | 3 | 1 | 0 |
| 3) Automata, Languages and Computation | 3 | 1 | 0 |
| 4) Language Processors | 3 | 1 | 3 |
| 5) Data Processing and File Structures | 3 | 1 | 3 |
| 6) Digital Electronics | 3 | 1 | 3 |
| | 17 | 6 | 9 |

| | | | |
|---|---|---|---|
| *SEMESTER VI* | | | |
| 1) Economics and Principles of Management | 2 | 1 | 0 |
| 2) Microprocessor Systems and Applications | 3 | 1 | 3 |
| 3) Principles of Programming Languages | 3 | 1 | 2 |
| 4) Operating Systems | 3 | 1 | 2 |
| 5) Data Communication | 3 | 1 | 0 |
| 6) Elective I | 3 | 0 | 3 |
| | 17 | 5 | 10 |

| | | | |
|---|---|---|---|
| *SEMESTER VII* | | | |
| 1) Open Elective I | 3 | 1 | 0 |
| 2) Algorithm Analysis and Design | 3 | 1 | 2 |
| 3) Information Systems Design | 3 | 1 | 2 |
| 4) Computers and Society | 3 | 1 | 0 |
| 5) Elective II | 3 | 0 | 3 |
| 6) Project | 0 | 0 | 3 |
| | 15 | 4 | 10 |

| | | | |
|---|---|---|---|
| *SEMESTER VIII* | | | |
| 1) Open Elective II | 3 | 1 | 0 |
| 2) Technical Writing and Communication | 2 | 1 | 0 |
| 3) Elective III | 3 | 0 | 3 |
| 4) Elective IV | 3 | 0 | 3 |
| 5) Project (continued) | 0 | 0 | 12 |
| | 11 | 2 | 18 |

## ACKNOWLEDGMENT

The model curriculum presented in this paper evolved through discussions of a large number of teachers and professionals and was organized by the Indian Society for Technical Education (ISTE). I would like to thank ISTE and all members who participated in the discussions leading to the final model curriculum.

## REFERENCES

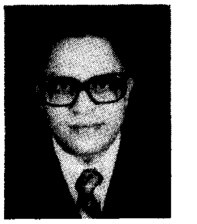
[1] "A modular curriculum in computer science," UNESCO-IFIP, UNESCO Publication, Paris, France, 1984.

[2] "Report of the panel on computer manpower development, electronics information and planning," Planning Commission Publication, New Delhi, India, vol. 8., pp. 55–70, Nov. 1980.

[3] "Model syllabi for four year degree course in computer science and engineering," Ministry of Human Resource Development, Government of India, New Delhi, Sept. 1987.

[4] Peter J. Denning *et al.* "Computing as a discipline," vol. 32, pp. 9–23, Jan. 1989.

[5] V. Rajaraman, "A curriculum proposal for a degree program in information system design," Computer Society of India Communications, pp. 5–8, Oct. 1981.



**Vaidyeswaran Rajaraman** received the B.Sc. (Hons.) in physics from Delhi University, India, in 1952, the Diploma and Associateship in electrical communication engineering from the Indian Institute of Science, Bangalore, India, in 1952 and 1955, respectively, the SM in electrical engineering from MIT, Cambridge, MA, and the Ph.D. degree from the University of Wisconsin, Madison, in 1961.

He was an Assistant Professor of Statistics at the University of Wisconsin, Madison, from 1961–1962. He joined Indian Institute of Technology (IIT), Kanpur, as an Assistant Professor of Electrical Engineering, in 1963, and became a Senior Professor of Electrical Engineering and Computer Science, in 1973. He headed the Computer Centre at IIT, Kanpur, from 1967–72 and again from 1976–79. He also initiated the computer science educational program and guided its growth from 1966–1979. Presently, he is a Professor of Computer Science and Chairman of Supercomputer Education and Research Centre at the Indian Institute of Science, Bangalore, India. He has published many technical papers and is the author of twelve books in computer science. From 1965–66, he was a visiting Assistant Professor of Computer Science and Electrical Engineering at the University of California, Berkeley, and from 1972–73 he was a Visiting IBM Research Fellow at the Systems Development Institute, Canberra, Australia. He has been active as a consultant to industry.

Dr. Rajaraman pioneered computer science education and research in India and in recognition of this he was awarded the Shanti Swarup Bhatnagar Prize in 1976 and the fellowship of the Computer Society of India in 1981. He was awarded the Homi Bhabha Prize for research in applied sciences by the University Grants Commission, India, in 1986, and a National Award for Excellence in Computer Engineering by the Indian Society for Technical Education, in 1988. He was a member of the Electronics Commission from 1979–1982 and is currently a Fellow of the Indian National Science Academy, Indian Academy of Sciences and Indian National Academy of Engineers, and an Honorary Professor of the Jawaharlal Nehru Centre for Advanced Scientific Research.

# Computer Operating Systems in Electrical Engineering Curriculum

Chyan Yang, *Senior Member, IEEE*

*Abstract*—**Diversified in their background knowledge, electrical engineering students cover a wide spectrum of strengths and weaknesses when they first attempt to study the subject of operating systems. Commonly, students are initially more concerned with the hardware design of digital systems and, therefore, less appreciative of the principles of software systems. In light of this, this paper presents and examines a practical syllabus for a course on operating systems. It also discusses possible derivations from this syllabus so as to provide versatility in implementation.**

## I. Introduction

When the earliest operational electronic computer provided the first services to mankind, with it came a set of operating procedures dedicated to the efficient utilization of the computer. First generation computers required complex procedures to maintain their operations. These procedures, though usually requiring some human interactions, were perhaps optimistically labeled as "operating systems." With advances in technologies and experiences in computer organizations, various computer architectures have evolved through the last three decades. The accompanying operating systems also have become more sophisticated. Many manual operating procedures are now automated and the need for human operations diminishes mainly to policy monitoring and file backups. Despite the changes, however, the issues concerning operating systems remain concrete. Although the relative importance of various issues fluctuates over time, the fundamental core of operating systems stems from the coordination of system resources and maximization of performance. Comparatively, electrical engineers design computers much in the same way that mechanical engineers design cars. We put functional components together to satisfy a hardware specification. The engineering efficiency of a car, however, is mainly determined by the entire construction or architecture of the car and not by the engine alone. Similarly, the performance of a computer relies not only on the hardware but also on the operating system that interfaces the hardware and the users. Many operating-system features may be the direct consequence of the underlying hardware support and many hardware specifications come directly from the operating system requirements. For instance, with indivisible instruction support, an operating system can provide concurrency control whereas an optimized compiler may call for a hardware design specification of register windows. The relationship between an operating system and its underlying hardware may sometimes appear as a chicken-and-egg problem: whether the hardware specification came from the operating system requirements or the operating system requirements came from the hardware reality? One could stake a convincing argument on either or both cases. In order to appreciate the wonder they set out to design, electrical engineering students must understand the importance and ramifications of operating systems.

## II. What to Learn and How Much?

It is not difficult to explain a CPU (central processing unit) to students of electrical engineering if they have had digital systems as their major. With a major in digital systems most students should have taken at least a course that uses the assembly language of some microprocessors. Instruction set architectures, macro definitions, and conventional branch instructions are terms covered in a standard syllabus of microprocessor programming. In addition, even students majoring in circuits and electronics, electromagnetics, control or communications commonly have knowledge of at least one high-level programming language. Be it as it may, not every student attending a class on operating systems is equipped with a proper background in basic computer organization. From experience, the understanding of the CPU functionality is a crucial part that cannot be skipped. Just as with an engine alone one cannot drive, with a lone CPU one cannot compute. An overall computer organization has to be reviewed. Hopefully, the functional components of a computer system can be covered within one lecture depending on the background of students. Only a few novices may have problems in this area.

Manuscript received July 1992.

The author is with the College of Management, National Chiao Tung University, Taiwan, Republic of China.

IEEE Log Number 9205768.