hardware and programming techniques, for realizing the linguistic features studied in the first course.

The subject “Information Systems” makes use of the material from the preceding subjects in studying the analysis, design and implementation of computer-based informations systems.

The second subject of this sequence is the key to inter-relating hardware and software concepts. A significant difference from conventional curricula is that students begin in their study of “Computation Structures” with a thorough understanding of the features found in a variety of source programming languages. With this background they are prepared to study the principles by which these features may be made available to computer users through the combination of hardware technology and programming concepts. An outline of this subject as it is currently taught at M.I.T. is given below.

**Computation structures**

1. **Logic Design:** Elementary combinational circuit synthesis; registers and gating; asynchronous modular systems (macromodules); sequential circuit synthesis; elementary implementation of arithmetic operations.

2. **Memory Systems:** Physical principles; name space—value space; distinction between location-addressed and associative memories; addressing by key transformation for associative retrieval.

3. **Computation Schemata:** Representation of a computation (in digital logic or as an abstract algorithm) by a set of operators that transform the contents of a set of memory cells. The domain and range cells of the operators are indicated by a data flow graph. The constraints that govern the sequencing of operator applications are specified by a precedence graph. Necessary and sufficient conditions for deterministic (unambiguous) operation are formulated. Extensions are made to represent procedures involving decision and iteration.

4. **Machine Organization:** Study of the principal forms of single-sequence processor organization and the corresponding techniques for compiling arithmetic assignments and conditional expressions: A simple single address machine; a stack-organized machine; machines having multiple general registers; machines having several functional units.

5. **Parallel Processing:** Multiprocess computer systems; process state, supervisor programs and scheduling; primitive procedure steps for representation of parallel computations: The fork, join and quit primitives; Dijkstra semaphores—the P(s) and V(s) operations. Process interlocking problems and their resolution.

6. **Nesting and Recursion:** Representation of an operator of one schema by a second schema. Naming of input and output quantities. Occurrence of multiple activations of a procedure through parallelism or recursion. Local data areas; use of stack storage allocation for single process implementation; base registers.

7. **Information Structures:** An information structure is modelled as a directed graph, without directed cycles, containing a directed path (not necessarily unique) to each node from a particular node called the root. Static operations on information structures; implementation by use of indexing. Dynamic operations; implementation by linked blocks in a location-addressed memory; garbage collection; implementation with associative memory.

---

**Let’s not discriminate against good work in design or experimentation**

*by GEORGE E. FORSYTHE*

*Stanford University*  
*Stanford, California*

I am distressed that graduate education in computer science is forcing students into a theoretical mold, and away from the vital practical problems of software engineering. I therefore urge that graduate computer science departments pay attention to the problems of experimentation and design in computer science. This might be done, for example, by employing faculty with interests in design and experimentation, by offering courses and examinations in these areas, and/or by accepting Ph.D. dissertations involving substantial designs or experiments of high quality. I believe that the last is the most important action to be taken now.

It seems to me that one main function of an educational system is to furnish society with imaginative performers and potential leaders in all the various areas of life. Within the computing field, there is a huge need for persons to create well designed, well documented software systems that exploit computers in the manifold ways we know to be possible. While the field will surely be advanced farthest by the creation of good new ideas, there remain enormous steps to be taken in exploiting...