COGNITIVE LEVEL DEMANDS OF TEST ITEMS IN
STATE-ADOPTED COMPUTER SCIENCE TEXTBOOKS

DISSERTATION

Presented to the Graduate Council of the
University of North Texas in Partial
Fulfillment of the Requirements

For the Degree of

DOCTOR OF PHILOSOPHY

By

James R. Aman, B.A., M.Ed.
Denton, Texas
August, 1988

Test items supplied with seven textbooks approved for use in Computer Science I and II curricula in Texas public schools were categorized by Bloom's taxonomy of educational objectives. Rating was done by a panel of ten judges selected from a group of participants at a taxonomy workshop. The selection criterion was demonstration of at least 80 percent competency in item classification. Judges received a small stipend for completing the rating task.

Of 2020 possible items, 998 were randomly selected for analysis. Equal percentages of items from each text were then randomly assigned to each rater. All statistical analyses were computed using SPSS/PC+ (version 2.1).

In both courses, CLD frequencies decreased through the three lower levels. The percentage of questions falling in these levels was approximately 83 percent for both courses. However, the higher-level course contained almost 10 percent more Knowledge level questions than did the lower course. At the higher taxonomic levels, the decline was roughly five percent per level in CS I but erratic in CS II. Analysis by book also revealed wide differences within each course.
Copyright by
James R. Aman
1988
ACKNOWLEDGEMENTS

There is no adequate way to say "Thanks" for all the support I have received at North Texas. Certainly the support of the University Doctoral Fellowship must rank high on the list, permitting me financial security and permitting a great deal of uninterrupted study time. I am grateful to the Graduate School for inviting me to be a participant in that program.

This study was funded in part by a grant from North Texas State University. Special thanks go to the staff of the Office of Research and Academic Grants for their constant help and encouragement.

My graduate student peers have been of a most important source of support. Together we have made everything work, helping each other through the rough spots and sharing the good moments. A simple acknowledgement of their contribution seems hardly enough.

Finally, I must acknowledge one special, absent friend for freely sharing thoughts and ideas, offering consolation, and often providing a little push. You helped far more than you ever realized! In my lifetime, I shall not be able to repay your kindness and support.
# TABLE OF CONTENTS

LIST OF TABLES ........................................... vii
LIST OF FIGURES ........................................ viii

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>I. INTRODUCTION TO THE STUDY</td>
<td>1</td>
</tr>
<tr>
<td>Theoretical Background</td>
<td>1</td>
</tr>
<tr>
<td>Textbook Adoption and Selection</td>
<td>4</td>
</tr>
<tr>
<td>Bloom's Taxonomy</td>
<td>6</td>
</tr>
<tr>
<td>Purpose of the Study</td>
<td>9</td>
</tr>
<tr>
<td>Research Question</td>
<td>10</td>
</tr>
<tr>
<td>Definitions of Terms</td>
<td>10</td>
</tr>
<tr>
<td>Abbreviations Used in the Study</td>
<td>11</td>
</tr>
<tr>
<td>Assumptions of the Study</td>
<td>11</td>
</tr>
<tr>
<td>Delimitations</td>
<td>12</td>
</tr>
<tr>
<td>II. SYNTHESIS OF RELATED LITERATURE</td>
<td>13</td>
</tr>
<tr>
<td>Thinking Skills</td>
<td>13</td>
</tr>
<tr>
<td>Bloom's Taxonomy of Educational Objectives</td>
<td>23</td>
</tr>
<tr>
<td>Educational Issues</td>
<td>24</td>
</tr>
<tr>
<td>Statistical Issues</td>
<td>26</td>
</tr>
<tr>
<td>III. METHODOLOGY OF THE STUDY</td>
<td>30</td>
</tr>
<tr>
<td>The Data</td>
<td>30</td>
</tr>
<tr>
<td>Item Selection</td>
<td>30</td>
</tr>
<tr>
<td>Rater Training and Selection</td>
<td>32</td>
</tr>
<tr>
<td>Assessment of Cognitive Level Demands</td>
<td>34</td>
</tr>
<tr>
<td>Data Analysis</td>
<td>34</td>
</tr>
</tbody>
</table>
## LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Test Bank Item Distributions</td>
<td>31</td>
</tr>
<tr>
<td>2</td>
<td>Percentage CLD Distribution: Computer Science I Textbooks</td>
<td>38</td>
</tr>
<tr>
<td>3</td>
<td>Range of CLD Percentage Frequencies: Computer Science I</td>
<td>39</td>
</tr>
<tr>
<td>4</td>
<td>Percentage CLD Distribution by Chapter: Mandell (CS I)</td>
<td>40</td>
</tr>
<tr>
<td>5</td>
<td>Percentage CLD Distribution by Chapter: Nance (CS I)</td>
<td>42</td>
</tr>
<tr>
<td>6</td>
<td>Percentage CLD Distribution by Chapter: Koffman (CS I)</td>
<td>43</td>
</tr>
<tr>
<td>7</td>
<td>Percentage CLD Distribution by Chapter: Lamb (CS I)</td>
<td>44</td>
</tr>
<tr>
<td>8</td>
<td>Percentage CLD Distribution: Computer Science II Textbooks</td>
<td>47</td>
</tr>
<tr>
<td>9</td>
<td>Range of CLD Percentage Frequencies: Computer Science II</td>
<td>48</td>
</tr>
<tr>
<td>10</td>
<td>Percentage CLD Distribution by Chapter: Graham (CS II)</td>
<td>50</td>
</tr>
<tr>
<td>11</td>
<td>Percentage CLD Distribution by Chapter: Naps/Singh (CS II)</td>
<td>51</td>
</tr>
<tr>
<td>12</td>
<td>Percentage CLD Distribution by Chapter: Peters (CS II)</td>
<td>52</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>1</td>
<td>Distribution of cognitive level demands for test items of Computer Science I textbooks</td>
<td>37</td>
</tr>
<tr>
<td>2</td>
<td>Distribution of cognitive level demands for test items of Computer Science II textbooks</td>
<td>46</td>
</tr>
</tbody>
</table>
CHAPTER I

INTRODUCTION TO THE STUDY

We want [our students] to know how to think for themselves, to respond to important questions, to solve problems, to pursue an argument, to defend a point of view, to understand its opposite, and to weight alternatives.

-- William J. Bennett (1987, p. 4)

Theoretical Background

With each passing day, computers advance further upon our daily lives. From industry to national defense to classrooms to our kitchens, bedrooms, and automobiles, computers in one form or another sometimes simplify and sometimes amplify our lives. The development of improved computer hardware with increasing capabilities has become a matter of fact. But every computer, regardless of size or capability, requires a human element--a programmer or team of programmers. The emergence of computer science courses in the secondary curriculum is partially a response to the increasing need for identifying and developing these individuals' talents and interests.

But before any student receives a computer science textbook, before a school district selects a book for the course, even before the State of Texas adopts the textbooks for use in the course, publishers and authors must create supplementary materials. In response to State requirements,
test banks are one of those additional items. However, the efficacy of any test depends upon the congruency of test questions and course objectives. The issue in such an assessment is the cognitive level demands (CLDs), that is, the taxonomic level at which each objective is written and at which it is tested.

However, the 1,100 school districts in Texas exercise local autonomy over specific course objectives. The sole common factor among objectives for any course is the State Board of Education Rules for Curriculum (1984). The Essential Elements, as these rules are commonly known, dictate overall course content. Administered through the Texas Education Agency, topical content is specified for the courses which may be taught in public schools.

The verbiage of the Essential Elements is broad. For computer science (see Appendix B) each of the fifteen sections is preceded by the directive that "the student shall be provided opportunities to: ...." The concern is not curriculum design but overall direction, content exposure rather than content mastery.

Public schools in Texas must deliver the curricular content specified in the Essential Elements. Approved textbooks, purchased by the State of Texas for distribution to school districts, must address the Essential Elements for their respective curricula. Texas is the single largest textbook purchaser in the United States. Because of the
state's favorable position as a textbook purchaser, Texas schools are assured of having texts which address state-specified curricular content.

Texas's economic leverage also places the state in the position of affecting curriculum nationwide. Books and supplementary materials are prepared with the Essential Elements in mind. Schools in other states have little choice but to adapt their curricula to fit the available textbooks. Supplemental materials, designed around the Texas Essential Elements, receive wide circulation. Diversifying test item CLDs has an importance beyond Texas's borders.

The Essential Elements for the two computer science courses require students to be exposed to increasingly complex algorithms and to continually develop programming skills. Planning any problem solution implicitly demands conceptualization of the problem at hand. The coding process itself is a minor task when compared with the careful planning and evaluation which must first occur. But the process of teaching a computer language, Pascal in this case, also involves a great deal of simple knowledge acquisition.

Computer Science I, the initial course in the sequence, focuses on the syntax and grammar of Pascal and on program design principles. In much the same way a child learns to use punctuation correctly in an English class, a neophyte
programmer must learn the grammatical rules of Pascal. Improper placement of a punctuation mark or the misuse of a key word can prevent a program from being properly read by the computer. A somewhat more sophisticated problem arises when the logical design of a program is faulty. In this case, however, execution occurs, but inappropriate or incorrect results may be obtained. Because the course content is fundamentally knowledge-gathering and recognition, course objectives and, consequently, testing might be expected to be at fairly low cognitive demand levels.

Computer Science II is concerned with creating, implementing, and manipulating various data structures. The analogy with an English class may be extended in this case to include sentence structure and phrase subordination. Program design focuses not on logic but on data structures. The cognitive demands are somewhat higher than in the earlier course, at least in theory.

Textbook Adoption and Selection

Textbooks used in Texas public schools are adopted by the State Board of Education and furnished to students without cost under provisions of the Texas Administrative Code (1985). Every six years, the State Board of Education issues a general proclamation advertising bids on certain textbooks. The current computer science adoption was the
state's first for the CS I and CS II curricula and occurred under Proclamation 62 (State Board of Education, 1985).

The procedures for adoption followed a strict schedule. Every step of the process had a deadline. The cycle for Proclamation 62 spanned the period from January, 1985, through May, 1987. Key steps included public hearings, the appointment of individuals to the State Textbook Committee, and various actions by publishers.

After the official proclamation was issued, publishers had 12 months to prepare statements of intention to bid specific books. Copies of books and supplemental materials had to be deposited with the Texas Education Agency and each Regional Education Service Center. These steps occurred during Spring, 1986.

Publishers' representatives visited State Textbook Committee members and made formal presentations before that committee during Summer, 1986. The actual selection of books occurred in late August, 1986. After revision, review, and additional public hearings, the final list of adopted textbooks was mailed to local school districts three months later. At that point, the focus of the selection process shifted to the district level.

Districts were permitted to select adopted textbooks and to place orders to a quota limit of 110% of subject enrollment. Multiple texts could be ordered within this quantity limit. Although procedures might vary across the
state, most districts would conduct their own assessments of the available texts.

As part of the textbook selection process, a school district might consider how well a test bank addresses the cognitive level demands of course objectives. Meeting Essential Elements content guidelines does not guarantee congruence per se. By determining test bank item CLDs, textbook evaluators would have a basis for assessing the relationship.

*Bloom's Taxonomy*

For over thirty years, Bloom's taxonomy of educational objectives (Bloom Engelhart, Furst, Hill, & Krathwohl, 1956) has been the accepted criterion for classifying intellectual educational behavior. Bloom's committee outlined a hierarchical taxonomy in the cognitive domain. The six basic categories were knowledge, comprehension, application, analysis, synthesis, and evaluation. In this order, the levels covered a continuum from simple recognition and factual recall (knowledge) through the complex process of making judgments (evaluation).

Knowledge is defined as recall of specifics and universals, of methods and processes, or of a pattern, structure, or setting (Bloom, Hastings, & Madaus, 1971). Students are required to remember facts already learned. No utilization of the information is required (Collette & Chiappetta, 1984). Knowledge is further divided into
knowledge of specifics, knowledge of ways and means of dealing with specifics, and knowledge of the universals and abstractions in a field.

Comprehension is the lowest level of understanding (Bloom et al., 1971). Students must organize, arrange, and interpret facts previously learned. They must be able to demonstrate the acquisition of sufficient background and understanding to use what has been learned (Collette & Chiappetta, 1984). Comprehension is subdivided into translation, interpretation, and extrapolation.

Application is defined as use of abstractions in particular and concrete situations (Bloom et al., 1971). At this level, students provide solutions based upon the identification of relevant information and rules (Collette & Chiappetta, 1984) or to restructure a problem (Wulf & Schave, 1984).

Analysis is the reasoning level (Chamberlain & Hardwick, 1986), calling for the separation of learning into constituent elements "such that the relative hierarchy of ideas is made clear and/or the relations between the ideas expressed are made explicit" (Bloom et al., 1971, p. 272). In other words, the relationships of parts to one another and to the whole are demonstrated (Wulf & Schave, 1984). Analysis has three subclasses: analysis of elements, analysis of relationships, and analysis of organizational principles.
Synthesis is the level of creativity (Chamberlain & Hardwick, 1986; Collette & Chiappetta, 1984). Elements and parts are assembled to form "a pattern or structure not clearly there before" (Bloom et al., 1971, p. 272). Phenomena must be analyzed, and hypotheses explaining them must be formulated. Bloom subdivided this class into production of a unique communication, production of a plan or proposed set of operations, and derivation of a set of abstract relations. Collette and Chiappetta (1984) make two important observations about synthesis level questioning. First, teaching objectives seldom call for student creativity. In well-constructed tests, therefore, synthesis questions are rarely used. Second, questions at this level will usually have a variety of plausible answers rather than a single correct answer.

Evaluation is the most complex category of Bloom's taxonomy. Students must render judgments about material and methods. This level is subdivided into judgments in terms of internal evidence and judgments in terms of external criteria (Bloom et al., 1971). Evaluation "implies that the learner can think logically about an issue in regard to specific reference points" (Wulf & Schave, 1984). The other five levels of the taxonomy are required to varying degrees in making appraisals. There will seldom be one correct answer to a question at this level, because of the diversity of assessment criteria which may exist.
Bloom's taxonomy has been particularly important to curriculum designers. By varying both the types and levels of objectives, mastery learning can be more effectively implemented. The limits of the learning process can be pressed within the framework of the classification hierarchy. Wulf and Schave (1984) contend that activities producing cognitive gain can be better balanced, leading to better individualization of instruction.

Purpose of the Study

By requiring the inclusion of "at least one test for each unit or chapter" in the teacher's edition or annotated edition of an adopted textbook (State Board of Education, 1985, p. 70), the use of supplied test banks was recognized as common practice by the State.

Course objectives are usually written at the school district level. Evaluation of the cognitive level demand (CLD) of the objectives properly occurs there, also. Matching a textbook with a course requires evaluation of the test bank CLDs. Because of the time and manpower required, this task may be prohibitive for many districts. Nonetheless, as a step in the textbook selection process, the information should be gathered and analyzed. The purpose of this study was to produce a description of the cognitive level demands of test items in state-adopted computer science textbooks.
Research Question

To carry out the purpose of this study, the following research question was offered: What are the distribution characteristics of the CLDs of test items in textbooks approved for the Texas computer science curricula?

Definitions of Terms

For the purpose of this study, the following operational definitions are used.

1. Cognitive level demand refers to the level of Bloom's taxonomy assigned to a test item by a trained, independent rater.

2. Bloom's taxonomy refers to the hierarchy of cognitive processes known fully as Bloom's taxonomy of educational objectives in the cognitive domain (Bloom et al., 1956).

3. Higher-order thinking skills are the upper three levels of Bloom's taxonomy: analysis, synthesis, and evaluation.


5. Computer Science I refers to a one-credit course of study outlined in section 75.124(a) of the Essential Elements. The emphases of this course are beginning level concepts and methods for command-level programming.
6. *Computer Science II* refers to a one-credit course of study outlined in section 75.124(b) of the Essential Elements. The emphases of this course are data structures, advanced concepts and methods, and algorithm development.

7. *Rater* refers to an individual assigning CLD values to test bank items in this study.

**Abbreviations Used in the Study**

1. *CLD* is an acronym for cognitive level demand, categories of Bloom's taxonomy of educational objectives in the cognitive domain.

2. *CS I* is an acronym for the Computer Science I course referenced above.

3. *CS II* is an acronym for the Computer Science II course referenced above.

4. *TEA* is an acronym for the Texas Education Agency.

**Assumptions of the Study**

1. State-approved textbooks for CS I and CS II meet or exceed the curricular requirements of their respective Essential Elements. This condition is a part of the state's textbook adoption process.

2. Raters judged test items solely on the basis of guidelines provided as a part of the training portion of this study.

3. The CLDs of test items can be validly and accurately determined.
Delimitations

1. The study did not attempt to suggest an appropriate distribution of cognitive level demands for test items in the computer science curricula.

2. The study was limited to the test banks accompanying textbooks under current adoption for the Texas computer science curricula.
CHAPTER II

SYNTHESIS OF RELATED LITERATURE

The purpose of this study was to produce a description of the cognitive level demands of test items in state-adopted computer science textbooks. Three areas of theoretical importance to the study will be reviewed: interest in the teaching of thinking skills, Bloom's taxonomy of cognitive educational objectives, and the development of educational objectives and test materials.

Thinking Skills

For thirty years national educational reformers have emphasized the teaching of critical thinking and problem solving skills in their proposals (Boyer, 1983; Bruner, 1961; Commission on the Humanities, 1980; Conant, 1959; Sizer, 1984). Adler (1982) considered them essentials of basic schooling. As activities for developing intellectual skills, "what is learned ... is skill in performance, not knowledge of facts and formulas" (Adler, 1982, p. 27). National commissions have been more specific.

The National Commission on the Humanities (1980) recommended that thinking skills be included with reading, writing, and arithmetic as a basic skill. Oxman (1984) suggested reconceptualizing the three traditional skills as
thinking skills themselves. The National Commission on Excellence in Education (1983) pointed to the absence of higher order intellectual skills among many 17-year-olds as one indicator of the risk facing the United States. In recommending content changes, the commission suggested that pre-high school curriculum provide a sound base for study in computational and problem solving skills, among other areas. The words of Secretary of Education William Bennett, used as the epigraph to this dissertation, summarize the reformers' motivation: "We want [our students] to know how to think for themselves, to respond to important questions, to solve problems, to pursue an argument, to defend a point of view, to understand its opposite, and to weight alternatives" (Bennett, 1987, p. 4). In a few short years, the computer has become the focus of these efforts.

Fostering critical thinking skills has occupied curriculum designers throughout modern history. Latin was the heart of the curriculum until quite recently (Upchurch & Lochhead, 1987). When transfer experiments discredited Latin as a discipliner of the mind, mathematics attempted to fill the role (Pressey, 1933). Again, the cognitive skills did not transfer (Upchurch & Lochhead, 1987). Today, computer programming is touted as the likely agent.

The possible impact on generalizable cognitive skills has been used as an argument in favor of the teaching of programming (Salomon and Perkins, 1987). The effect of
this training could be quite powerful. Papert (1980), inventor of the Logo language, wrote

The child programs the computer. And in teaching the computer how to think, children embark on an exploration about how they themselves think. ... Thinking about thinking turns the child into an epistemologist. (p. 19)

In spite of some negative opinions about transferability (Clements, 1985; Dalbey & Linn, 1985; Pea & Kurland, 1983), efforts continue to be made. Thinking skills continue to be high on the agenda for curriculum specialists.

Current educational literature contains frequent reports related to the teaching of thinking skills. Psychoeducational knowledge has advanced sufficiently that success in the effort is a promising prospect (Quellmalz, 1985; Sternberg, 1985). There is a "feeling among educators that, in trying to make students better thinkers, we have tried pretty much everything else to no avail, so that the time to teach critical thinking directly is surely at hand" (Sternberg, 1985, p. 194). When the school environment fosters initiative and independence, thinking as a process flourishes (Feldman, 1986).

The thinking process begins with creativity--brainstorming possible solutions to a problem. The transition to critical thinking occurs as the focus narrows and alternatives are evaluated. A trend in professional training provides a contemporary example.
Medical education has traditionally been a content-intensive field. In 1968 McMaster University introduced a problem-based curriculum emphasizing critical thinking rather than memorization (Abrahamson, 1987). Medical schools in Israel, Australia, and the Netherlands followed suit over the next decade. Medical schools at Harvard, Mercer, Rush, Tufts, and Wake Forest Universities, and at the University of New Mexico have implemented problem-based curricula in the 1980’s. This innovation has stirred a great deal of discussion, although the concept is a simple one.

Students meet in small tutorials and work on biomedical problems demanding the acquisition of new information and skills. Through a carefully designed hierarchy of case studies, "their knowledge and understanding of basic and clinical sciences develop systematically" (Abrahamson, 1987, p. B1). Abrahamson reported enthusiastic response to the curriculum among faculty and students. More important, students’ problem-solving abilities and intellectual curiosity were heightened beyond what was observed in more traditional medical schools.

Efforts to introduce the teaching of thinking into the precollege curriculum have proliferated in the past few years. Costa noted the difficulty of finding statistical verification of effectiveness; most research evidence is qualitative (Brandt, 1988). Hudgins and Edelman (1986)
studied critical thinking skills development in fourth and fifth graders. Following small group discussions, a critical thinking test was administered and evaluated. Students talked more—and teachers less—by the end of the study. Students were providing more evidence and offering more conclusions, which was interpreted as an increase in critical thinking.

Empirical research on the use of the "Feurstein Instrumental Enrichment" (FIE) materials as a method of teaching thinking skills was reviewed by Savell, Twohig, and Rachford (1986). The authors identified the two components of the FIE technique as

(a) a set of 14 paper-and-pencil exercises designed to help students identify basic principles of thinking and to practice self-monitoring with respect to these principles, and (b) a set of training procedures involving teacher-guided "bridging" back and forth between the principles identified in the exercises and various subject matters of interest. (p. 382)

The studies had been conducted in Israel, Venezuela, the United States, Canada, and with several special populations. Some results were difficult to interpret; others reported no clear FIE effects.

The negative effects of the teaching of higher-order thinking skills received attention from Lohman (1986). In reviewing research showing dysfunctional effects for teaching thinking, Lohman identified two types of cognitive skills (domain-independent and domain-specific) as organized
sets of production rules. The effects of training efforts may be predicted from existing procedural knowledge. Fluid and crystallized abilities require different types of materials.

Inservice training has also received recent attention. Although not empirical in nature, recent articles have emphasized the importance of teacher preparation. The importance of trained teachers was noted in programs specifically for at-risk students (Pogrow, 1988). Tabor (1988) described the reactions of teachers in Irvine, California, to a local thinking skills effort. The primary emphasis of the Irvine Thinking Project was “on a specific set of teacher verbal behaviors rather than on direct instruction in thinking skills” (Tabor, 1988, p. 49). Peer coaching and observation were the inservice vehicles.

Another inservice model was specifically designed for teachers of gifted children (Schlichter, 1988). Using the Talents Unlimited Inservice Education Model, regular classroom teachers and specialists in gifted education were trained. Identification of students "with high potential in a wide range of abilities" (Schlichter, 1988, p. 122) and enrichment in areas of strength were seen as outcomes of the Talents Unlimited program.

Numerous suggestions for formalizing the teaching of thinking skills as a general topic have also been made (Bratton, 1988; Chambers, 1988; Mirman & Tishman; 1988;
Beyer (1988) emphasized the importance of a well-structured thinking skills scope and sequence as a framework "not overwhelmed by academic subject matter" (p. 26). The importance of teaching both content and process was pointed out by Presseisen (1988).

Nonetheless, the teaching of critical thinking as a skill in and of itself continues to draw criticism. Baer (1988) expressed skepticism regarding the effect of such teaching on better students. "We must appreciate that good thinking is in no way less valuable for being produced in unconventional (or unconscious) ways" (Baer, 1988, p. 69). The product, not the process, should be considered as most important.

In rebuttal, Perkins (1988) suggested Baer had been overreacting. Almost all students, at some time and at whatever level, think poorly. Thinking instruction would benefit most students. Baer's wariness was downplayed as a scare tactic.

Adler (1987) flatly contends the practice is not possible. Courses designed specifically to teach such skills are considered ill-conceived and doomed to failure. Teaching which does not require students to think about what is being taught merely constitutes indoctrination.

Boyer (1983) enumerated five factors explaining the current lack of success in teaching thinking skills. First, doubt about which skills actually comprise "thinking skills"
has left educators confused. Second, when a decision is
finally made about which skills to teach, educators fail to
identify appropriate components. Third, inappropriate
teaching techniques are employed. Fourth, thinking skills
curricula attempt to cover too many skills in too little
time. Fifth, what is taught and what is tested as thinking
skills are dissimilar.

Sternberg (1985) dealt directly with the issue of
teaching critical thinking. The difficulty of transferring
thinking skills across domains was emphasized. To provide a
bridge, current training efforts should be supplemented with
"training that involves solving real-life problems"
(Sternberg, 1985, p. 277), the thrust of a program called
Intelligence Applied. Problems with clearcut answers are
mixed with practical problems lacking absolute solutions.

If we wish to prepare students to solve the problems
they will confront in their lives, then we must present
them with realistic simulations of real problems, not
merely with problems that are tailored to our
convenience because they are objectively scorable or
have been removed from context. (Sternberg, 1985, p.
278)

This sentiment is representative of the conflict
between teaching content and teaching process. Content
usually takes precedence. One solution is to use the
computer to create "a manageable environment in which the
focus can be placed on process" (van Deusen & Donham, 1986-
87, p. 32). Many computer programs are available today to
assist in teaching thinking skills, primarily because
software manufacturers have identified thinking skills as a curricular niche for the computer.

The literature regarding computer use in the development of thinking skills is focused heavily on programs which may be useful. Mrosla (1988), for example, built a case for using such software but concentrated on descriptions of program types with some specific examples. King (1988) identified five programs for use in computer mathematics courses. Others have followed similar courses in other subject areas (Blubaugh, 1987; Coburn, 1988; Cochran, 1988; Fox, 1987; Fritz & Weaver, 1986; Mattarocci & Rasa, 1988).

Specific guidance for incorporating software into the curriculum has been offered (Billstein, 1987-88; Crane, 1988; Duke, 1988; Rooze, 1988; Rooze & Northup, 1986; van Deusen & Donham, 1986-87). Notably absent from the scene is organized pedagogical research on the use of computers in the teaching of problem solving and critical thinking skills. Laudable as published case studies may be, they are only a starting point. Formal research efforts should be made to test and evaluate computer-assisted curricula in this area.

But the software is only the teaching medium. The teacher continues to be the facilitator, the controller of the learning process. In the final analysis, the computer can only help create an environment conducive to learning.
The teacher must perform the critical step of affecting a transfer of learning from the sheltered computer environment to other situations (van Deusen & Donham, 1986-87).

Because of cognitive similarities between the fields, numerous studies have drawn parallels between coursework in computers and mathematics (Fertsch, 1985; Griswold, 1983; Hannafin & Cole, 1983; Loyd & Gressard, 1984; Swadener & Hannafin, 1987). Both areas focus attention on algorithm development and application. But a major problem in mathematics instruction has been a frequent lack of originality in algorithmic skills. What teachers demonstrate, students can simply parrot. Considering or offering alternative solutions has been discouraged (Hatfield, 1982). Programming provides a sharp contrast. Many cognitive processes may be required to develop a computer program's algorithm, with secondary benefits accruing in the form inherent mathematical concepts (Hatfield, 1982).

To be effective in teaching higher order cognitive skills, a programming course must go beyond merely teaching the mechanics of a language. Upchurch and Lochhead (1987) warn of unpredictable educational outcomes when computer instruction is not built around problem solving. Evidence has been offered that learning to program is not sufficient to improve thinking ability or problem-solving skills (Pea & Kurland, 1983). However, transfer of learning is not an
isolated occurrence. Specific reference should be made to the application of ideas from programming to other subject areas (Upchurch & Lochhead, 1987).

Bloom's Taxonomy of Educational Objectives

When Bloom's committee suggested a hierarchical taxonomy of learning for the cognitive domain over thirty years ago, they touched off a firestorm (Bloom et al., 1956). The taxonomy envisioned by Bloom contained six major classes: Knowledge, Comprehension, Application, Analysis, Synthesis, and Evaluation. Some classes were further subdivided. Ordering educational objectives from simple to complex, the group sought to remove what they considered rigidity in teaching in the early days of the mastery learning movement. In spite of philosophical dissension, the taxonomy has gained widespread acceptance (Furst, 1981).

Bloom's taxonomy "is a classification of student behavior and as such is independent of the subject matter which is being studied" (Bloom, 1956, p. 206). Likewise, complexity of subject matter does not seem to be of importance. In a study of test items from a dental school's predoctoral written examination, roughly 84 percent were at the Knowledge level and the remainder at the Comprehension level (Rinchuse & Zullo, 1986).

Bloom's taxonomy is built on two basic assumptions: first, that the behaviors at each level constitute a behavioral hierarchy; and, second, that the statistical
concept of item difficulty is equivalent to the psychological concept of item difficulty or, perhaps more accurately, item complexity" (Poole, 1971, p. 379). The psychoeducational and statistical issues raised have been investigated repeatedly.

**Educational Issues**

Bloom's committee attempted to reflect distinctions between student behaviors already being made by teachers (Bloom et al., 1956). The taxonomy was not presented as permitting complete and sharp distinctions among those behaviors. Students observably worked the same problem in different ways. A test item might be classifiable under multiple levels depending on prior learning experiences.

Bloom acknowledged this problem, considering knowledge or assumptions about students' prior learning experiences essential before any assignment of items or objectives to different categories (Bloom et al., 1956). This stipulation has also been used to argue against the use of behavioral-specified goals (Furst, 1981). Nonetheless, Bloom felt assessment of the taxonomy should be based on the degree to which educators could agree on classifying objectives or items into distinct classes (Seddon, 1978).

Bloom and his co-workers reported a series of classification studies. They gave no details, stating only that
although we have little difficulty in determining the major class within which a behavior falls, we are still not satisfied that there are enough clearly defined subclassifications to provide adequately for the great variety of objectives we have attempted to classify. (Bloom et al., 1956, p. 21)

However, other studies have yielded mixed results. Using only main categories, Scannell and Stellwagen (1960) achieved 90 percent agreement with two judges. Stoker and Kropp (1964) conducted two studies using four judges; they reported 31 percent agreement. Poole conducted two studies. In the first, six judges attained 16 percent agreement (Poole, 1971). In the other, seven judges agreed on 14 percent of the items (Poole, 1972). Fairbrother (1975) used 22 judges in each of two studies of 40 items, reporting only seven percent agreement on one and no agreement on the other. Other studies found agreement ranging from 40 percent to 75 percent (Herron, 1966; Cox, 1965; Tyler, 1966). Agreement was inversely related to the number of judges across these studies. Seddon (1978) attributed this phenomenon to the potential increase in disagreements between pairs of judges.

Could judges reliably assign taxonomic levels? Reports provided no consensus. Scannell and Stellwagen (1960) felt the accuracy they obtained was reasonable. Stoker and Kropp (1964) were similarly--and cautiously--positive. Stanley and Bolton (1957) endorsed "regular analysis of teacher-made and standardized tests" (p. 634). Poor results obtained by
Poole (1971, 1972) and Fairbrother (1975) led both to reject attempts at item classification.

Furst (1981) considered this a moot issue. Bloom had described the taxonomy as a purely descriptive scheme. Goals could be represented in relatively neutral ways (Bloom et al., 1956). Educational objectives which could not readily be specified as changes in student behavior were not acceptable. Furst claimed this violated the avowed neutrality, stating "Classifications tend to throw emphasis on certain qualities and, in turn, to diminish the apparent significance of other qualities" (p. 442). Whether or not judges could rate reliably did not matter in the final analysis.

Statistical Issues

Bloom's taxonomic hierarchy was intended to be cumulative. For a narrow subject-matter area, the taxonomy would constitute a simplex, a term referring to a set of variables differing only in terms of degree of complexity rather than content (Guttman, 1953). In broader contexts variables would differ by both content and complexity. This condition was called a radex by Guttman (1953).

The major levels of the taxonomy were expressed by Hill and McGaw (1981) as a path model with paths existing only between adjacent levels: Knowledge (K) -> Comprehension (C) -> Application (Ap) -> Analysis (An) -> Synthesis (S) -> Evaluation (E). Numerous attempts to test the simplex
assumption have produced no clear picture. The seminal study in the area was conducted by Kropp and Stoker (1966). Their data have been reanalyzed in whole or part on several occasions.

The original study involved high school students in ten Florida schools. Four tests of 95 items each were administered to 1100-1500 students in each grade. Each test contained twenty questions in the Knowledge, Comprehension, Application, and Analysis categories, five in the Synthesis category, and ten in the Evaluation category. Correlation matrices obtained between six subtests were analyzed using Kaiser's (1962) least squares procedure. Only one test produced the taxonomic order exactly at all grade levels. However, across all grades and tests, the order was maintained through the Analysis category. The research team decided that the consistent misplacement of Synthesis and Evaluation categories was due to faulty test items (Kropp & Stoker, 1966; Seddon, 1978).

Using causal modeling, another team of researchers reanalyzed a subsample of the Kropp and Stoker data (Madaus, Woods, & Nuttall, 1973). They reported several paths between nonadjacent levels (Hill & McGaw, 1981). They claimed the data supported a "Y"-shaped structure for Bloom's taxonomy. The stem extended from Knowledge to Application (K -> Ap) with one branch toward Analysis (Ap ->
An) and the other toward Synthesis and Evaluation (Ap $\rightarrow$ S $\rightarrow$ E) (Madaus et al., 1973).

Reinterpretation has not cleared the picture at all. A study by O'Hara, Snowman, and Miller (1978) disputed the methodology used to obtain the "Y"-shaped structure. They rejected structure and accepted the cumulative hierarchy with one additional non-adjacent path. But the following year, the same research team published a reanalysis of a subsample of the Kropp and Stoker data. Using path analysis, stepwise regression, and factor analysis, they concluded that the simple hierarchy could not be supported. Path analysis had led them back to a Y-shaped structure (Miller, Snowman, & O'Hara, 1979).

Using a modeling method for the analysis of linear structural relationships (LISREL), Hill and McGaw (1981) reanalyzed both the Kropp and Stoker and O'Hara et al. data. Their results suggested the separation of Knowledge from the remainder of the hierarchy. Following a suggestion by Madaus et al. (1973), a general ability or intelligence factor was also introduced. "The reduced five-factor model, eliminating both Intelligence and Knowledge, was considered to provide the best explanation of the data" (Hill & McGaw, 1981, p. 99).

Although no consensus has been reached, the weight of practice clearly favors the six-level hierarchical model. With further investigation over wider ranges of content and
conditions, an alternative may be identified, perhaps even accepted. In the meantime, the potentially important works of Hill and McGaw, Madaus et al., and O’Hara et al. will only serve to fuel the ongoing debate over Bloom’s taxonomy.
CHAPTER III

METHODOLOGY OF THE STUDY

The purpose of this study was to produce a description of the cognitive level demands of test items in state-adopted computer science textbooks. This study was descriptive and exploratory in nature.

The Data

The data for this study were taxonomic levels assigned independently by a group of raters to 998 test items drawn randomly from materials accompanying textbooks for CS I and CS II (see Appendix A). Four books from two publishers were adopted for CS I, containing a total of 1545 test items. Four books from three publishers were adopted for CS II, containing an additional 475 test items.

Item Selection

The original pool of test items included the full test banks from the adopted textbooks. For two reasons, no limitation was placed on question type. First, to limit the study to one type of question (for instance, multiple choice) would have inherently denied the legitimacy of other types as measurement tools. Second, if the items finally selected for evaluation were to be representative of their
<table>
<thead>
<tr>
<th>Book</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>n</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mandell</td>
<td>38</td>
<td>38</td>
<td>38</td>
<td>38</td>
<td>38</td>
<td>38</td>
<td>38</td>
<td>38</td>
<td>40</td>
<td>40</td>
<td>384</td>
</tr>
<tr>
<td>Nance</td>
<td>30</td>
<td>30</td>
<td>30</td>
<td>30</td>
<td>31</td>
<td>31</td>
<td>30</td>
<td>30</td>
<td>30</td>
<td>30</td>
<td>302</td>
</tr>
<tr>
<td>Koffman</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>6</td>
<td>8</td>
<td>78</td>
<td></td>
</tr>
<tr>
<td>Lamb</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>17</td>
</tr>
<tr>
<td>Graham</td>
<td>5</td>
<td>6</td>
<td>5</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>58</td>
</tr>
<tr>
<td>Naps &amp; Singh</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>8</td>
<td>7</td>
<td>75</td>
</tr>
<tr>
<td>Peters</td>
<td>9</td>
<td>9</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>9</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>84</td>
</tr>
<tr>
<td>n</td>
<td>99</td>
<td>100</td>
<td>98</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>101</td>
<td></td>
</tr>
</tbody>
</table>

respective textbooks, all test items must have had an equal chance of selection.

All 2020 test items were coded for identification. Using a stream of randomly generated numbers, fifty percent of the items were selected from each test bank with two exceptions. One text in CS I contained only 17 items; all of them were used. A text in CS II had no accompanying test bank and was not represented in the study. Because of the randomization, the final pool of 998 items was proportionally distributed by source text across the raters.
(see Table 1). Rater bias, which could have been a factor were each rater to have evaluated a single text, was eliminated by distributing the test banks across the entire rater pool.

**Rater Training and Selection**

Interrater reliability was established a priori. A pool of 13 trainees was recruited from the general population of the University of North Texas. These people were faculty, staff, graduate students, and upperclass undergraduate students. Instruction and practice in the identification and assignment of taxonomic levels was provided in a two-hour workshop by a specialist in educational measurement.

The workshop was built around presentations and discussions of Bloom's cognitive taxonomy. After an introduction to each level from a conceptual standpoint, keywords typically associated with the level were presented. Trainees then worked in small groups to assign levels to test items. Responses were discussed as a means of providing feedback and reinforcing the training.

At the end of the workshop, each trainee evaluated a bank of fifty test items of known characteristics. Ten raters were selected for participation in the study on the basis of performance in the evaluation exercise. The selection criterion was correct identification of taxonomic levels of at least 80 percent of the test items. The
selected rating group consisted of one faculty member, six graduate students, and three undergraduates. They were informed after their selection that an honorarium would be paid for completing the rating task.

Because the various test banks were in different typesets and of different sizes, each evaluation set was retyped. A bubble format was provided at the left margin with column headers for the six levels of Bloom's taxonomy. The consistent typography improved item readability. Providing the marking space with the question minimized the possibility of referencing the wrong item.

The raters reported a broad range of exposure to computer programming. Those with minimal experience were expected to have more difficulty interpreting some programming jargon. Everyone was provided with the following explanations for four terms which appeared often in the evaluation sets:

**Compiler or syntax error:** occurs when a command has been misspelled or when a required word or punctuation mark has been omitted. These errors are purely mechanical, arising from failure to apply the rules of the programming language.

**Run-time error:** caused by non-mechanical factors, such as dividing by zero. This type of error cannot usually be identified by simply looking at the program code.
Logic error: caused by a flaw in the program's design. The program will execute, but the results or the progress of the program will be unexpected.

Declare a variable or data type: a straightforward and "mechanical" process for telling the computer the kind of data associated with a particular variable or data name.

Assessment of Cognitive Level Demands

Each evaluation set contained identical directions. Raters were instructed to assign one taxonomic level per item. In a closing statement, raters were asked to check that there were no cases of double rating and that each item had received an assignment.

In some cases the original question referred to an accompanying program segment, flowchart, or diagram. That material was not considered pertinent to assessing a CLD. However, the rater's evaluation item contained a note indicating the original presence of the material.

Data Analysis

Because Bloom's taxonomy was hierarchical in nature, the CLD frequency data constituted an ordinal variable. Statistical analyses were limited to procedures appropriate for ordered data. Central tendency was described by medians and modes; dispersion was indicated by calculations of
frequency ranges. Skewness was computed as an indicator of shape. The data were also used to prepare histograms for each textbook and academic course. All computations were run under SPSS/PC+ (version 2.1), the microcomputer version of the Statistical Package for the Social Sciences (Norusis, 1986).

Limitations of the Study

Several factors limit the applicability of this study.

1. Training focused on key words not clearly applicable to the subject matter. The possibility of misassignment is increased under these conditions.

2. Only test banks supplementing state-adopted textbooks were analyzed. No attempt should be made to extend the findings beyond these seven sets.

3. Some raters had little computer experience. By their own admission, this affected their ability to interpret some test items.
CHAPTER IV

PRESENTATION OF THE DATA

The purpose of this study was to produce a description of the cognitive level demands of test items in state-adopted computer science textbooks. Ten raters assigned levels of Bloom’s taxonomy to a total of 998 items. Test items were selected randomly from supplemental tests for textbooks in both Computer Science I and Computer Science II. The research question was answered by inspecting the data by course and by book.

Computer Science I

Slightly more than one third of the test items examined for Computer Science I were judged to be at Knowledge level. This observation held for all textbooks except the Lamb text, which had only 17 test items. Percentages of items decreased steadily as taxonomic level increased (see Figure 1). Ninety percent of the questions were at or below the level of Analysis. In the individual textbooks, however, the percentages were widely varied.

The greatest concentration of questions (54%) in the Mandell text was at the Knowledge level (Table 2). Comprehension contained another 23 percent. Application and Analysis provided another 21 percent. Only two items were
Figure 1. Distribution of cognitive level demands for test items of Computer Science I textbooks.

Questions from the Nance text were evenly distributed at the three lower levels (Table 2). Over 75 percent of the judged test items were in this range. Both the Comprehension and Application categories had higher
Table 2

Percentage CLD Distribution: Computer Science I Textbooks

<table>
<thead>
<tr>
<th>CLD</th>
<th>Mandell</th>
<th>Nance</th>
<th>Koffman</th>
<th>Lamb</th>
<th>%</th>
<th>n</th>
</tr>
</thead>
<tbody>
<tr>
<td>Knowledge</td>
<td>53.9</td>
<td>24.8</td>
<td>23.1</td>
<td>0</td>
<td>38.4</td>
<td>300</td>
</tr>
<tr>
<td>Comprehension</td>
<td>23.2</td>
<td>27.8</td>
<td>16.7</td>
<td>23.5</td>
<td>24.3</td>
<td>190</td>
</tr>
<tr>
<td>Application</td>
<td>13.0</td>
<td>26.5</td>
<td>13.2</td>
<td>17.6</td>
<td>20.2</td>
<td>158</td>
</tr>
<tr>
<td>Analysis</td>
<td>9.4</td>
<td>11.6</td>
<td>9.0</td>
<td>29.4</td>
<td>10.6</td>
<td>83</td>
</tr>
<tr>
<td>Synthesis</td>
<td>0</td>
<td>8.3</td>
<td>17.9</td>
<td>29.4</td>
<td>5.6</td>
<td>44</td>
</tr>
<tr>
<td>Evaluation</td>
<td>0.5</td>
<td>1.0</td>
<td>1.3</td>
<td>0</td>
<td>0.8</td>
<td>6</td>
</tr>
<tr>
<td>n</td>
<td>384</td>
<td>302</td>
<td>78</td>
<td>17</td>
<td>100.0</td>
<td>781</td>
</tr>
</tbody>
</table>

percentages of questions than were found for Knowledge.
Decline at higher levels was more gradual than for other Computer Science I texts.

The Koffman text questions were unevenly distributed (Table 2). More items were judged to be Knowledge class than any other; the Comprehension and Application classes were roughly equal. However, the higher taxonomic levels did not follow the overall trend. A higher percentage of items was judged to be Synthesis than at any other level except Knowledge.

The Lamb text (see Table 2) was unique both in the total number of test items (17) and in the nature of the
Table 3

Range of CLD Percentage Frequencies: Computer Science I

<table>
<thead>
<tr>
<th>CLD Group</th>
<th>Low (K -&gt; Ap)</th>
<th>High (An -&gt; Ev)</th>
<th>Full (K -&gt; Ev)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Textbook</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mandell</td>
<td>50.9</td>
<td>9.4</td>
<td>53.9</td>
</tr>
<tr>
<td>Nance</td>
<td>3.0</td>
<td>10.6</td>
<td>26.8</td>
</tr>
<tr>
<td>Koffman</td>
<td>9.9</td>
<td>16.6</td>
<td>21.8</td>
</tr>
<tr>
<td>Lamb</td>
<td>17.6</td>
<td>29.4</td>
<td>29.4</td>
</tr>
</tbody>
</table>

Most chapters had only a single evaluation item, usually requiring creation of a full program. The observed range (12%) represented a difference of only two items. Nonetheless, the highest percentage of items (29% each) was found at the Analysis and Synthesis levels.

The overall distribution for CS I test items presented in Figure 1 demonstrated a steady decline in frequencies through the taxonomic levels. Eighty-three percent of the items were at lower cognitive levels, half at the Knowledge level. However, individual textbooks varied sharply. The ranges of CLD frequencies at lower and upper cognitive levels provided an indication of this characteristic.

As shown on Table 3, the Nance text had the most balanced distribution through the lower levels and only a 10
Table 4  

Percentage CLD Distribution by Chapter: Mandell (CS I)

<table>
<thead>
<tr>
<th>Chapter Topic</th>
<th>Kn</th>
<th>Co</th>
<th>Ap</th>
<th>An</th>
<th>Sy</th>
<th>Ev</th>
<th>n</th>
</tr>
</thead>
<tbody>
<tr>
<td>1: General Intro</td>
<td>85.0</td>
<td>10.0</td>
<td>5.0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20</td>
</tr>
<tr>
<td>2: Prgm. Methods</td>
<td>25.0</td>
<td>30.0</td>
<td>20.0</td>
<td>5.0</td>
<td>0</td>
<td>0</td>
<td>20</td>
</tr>
<tr>
<td>3: Data Types</td>
<td>65.0</td>
<td>30.0</td>
<td>5.0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20</td>
</tr>
<tr>
<td>4: Input/Output</td>
<td>44.4</td>
<td>11.1</td>
<td>38.9</td>
<td>6.6</td>
<td>0</td>
<td>0</td>
<td>18</td>
</tr>
<tr>
<td>5: Assignments, Math, If/Then</td>
<td>50.0</td>
<td>18.8</td>
<td>33.3</td>
<td>18.8</td>
<td>0</td>
<td>0</td>
<td>16</td>
</tr>
<tr>
<td>6: Loops &amp; Case</td>
<td>50.0</td>
<td>28.6</td>
<td>14.3</td>
<td>7.1</td>
<td>0</td>
<td>0</td>
<td>14</td>
</tr>
<tr>
<td>7: Loop Control</td>
<td>51.7</td>
<td>27.6</td>
<td>13.8</td>
<td>6.9</td>
<td>0</td>
<td>0</td>
<td>29</td>
</tr>
<tr>
<td>8: Style</td>
<td>42.1</td>
<td>26.3</td>
<td>5.3</td>
<td>26.3</td>
<td>0</td>
<td>0</td>
<td>19</td>
</tr>
<tr>
<td>9: Func. &amp; Proc.</td>
<td>63.0</td>
<td>14.8</td>
<td>18.5</td>
<td>3.7</td>
<td>0</td>
<td>0</td>
<td>27</td>
</tr>
<tr>
<td>10: Predef. Func. &amp; Proc.</td>
<td>41.2</td>
<td>35.3</td>
<td>17.6</td>
<td>5.9</td>
<td>0</td>
<td>0</td>
<td>17</td>
</tr>
<tr>
<td>11: Arrays</td>
<td>41.2</td>
<td>41.2</td>
<td>5.9</td>
<td>11.8</td>
<td>0</td>
<td>0</td>
<td>17</td>
</tr>
<tr>
<td>12: Records &amp; Sets</td>
<td>66.7</td>
<td>0</td>
<td>16.7</td>
<td>16.7</td>
<td>0</td>
<td>0</td>
<td>18</td>
</tr>
<tr>
<td>13: Sort &amp; Search</td>
<td>38.9</td>
<td>27.8</td>
<td>16.7</td>
<td>5.6</td>
<td>0</td>
<td>11.1</td>
<td>18</td>
</tr>
<tr>
<td>14: Files</td>
<td>50.0</td>
<td>35.7</td>
<td>7.1</td>
<td>7.1</td>
<td>0</td>
<td>0</td>
<td>28</td>
</tr>
<tr>
<td>15: Pointers, Lists</td>
<td>50.0</td>
<td>18.2</td>
<td>22.7</td>
<td>9.1</td>
<td>0</td>
<td>0</td>
<td>22</td>
</tr>
<tr>
<td>16: Dynamic Struc.</td>
<td>50.0</td>
<td>30.0</td>
<td>5.0</td>
<td>15.0</td>
<td>0</td>
<td>0</td>
<td>20</td>
</tr>
<tr>
<td>17: Numer. Algor.</td>
<td>76.0</td>
<td>16.0</td>
<td>8.0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>25</td>
</tr>
<tr>
<td>18: Comp. Appl'ns</td>
<td>64.3</td>
<td>21.4</td>
<td>7.1</td>
<td>7.1</td>
<td>0</td>
<td>0</td>
<td>14</td>
</tr>
<tr>
<td>19: Social Issues</td>
<td>59.1</td>
<td>18.2</td>
<td>9.1</td>
<td>13.6</td>
<td>0</td>
<td>0</td>
<td>22</td>
</tr>
</tbody>
</table>
percent range in the higher categories. The Mandell text had the greatest range at higher levels, an artifact of the large quantity of items at the Knowledge level. Over the full span of CLDs, frequencies were narrowest for the Koffman text.

Ultimately, the value of a taxonomic evaluation of these test banks lies in the degree to which the CLDs of test items match the CLDs of specific course objectives. A more detailed description of each textbook is required. Descriptions of the CS I test banks, divided by textbook chapter, are contained in Tables 4 through 7. The general topic of each chapter is provided along with the distribution of CLDs assigned by raters to questions from the respective tests.

The heavy concentration of items at lower levels in the Mandell text was immediately apparent (Table 4). Many of the topics in CS I are more conducive to treatment at lower cognitive levels. But sorting, searching, dynamic structures, numerical algorithms, and social issues are topics with higher level themes. However, the questions for this text used skills beyond Analysis level only for the chapter on searching and sorting. With 76 percent of the items at Knowledge level, the chapter on numerical algorithms seemed particularly skewed.

The even distribution of the Nance text has already been mentioned. Examination of the chapter distribution of
Table 5

Percentage CLD Distribution by Chapter: Nance (CS I)

<table>
<thead>
<tr>
<th>Chapter Topic</th>
<th>Cognitive Level</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Kn</td>
</tr>
<tr>
<td>1: General Intro</td>
<td>85.7</td>
</tr>
<tr>
<td>2: Simple Data Types, Math, Structures</td>
<td>11.8</td>
</tr>
<tr>
<td>3: VAR, CONST, I/O, Functions</td>
<td>10.7</td>
</tr>
<tr>
<td>4: Style, Debugging</td>
<td>25.0</td>
</tr>
<tr>
<td>5: Cond'il Stmts</td>
<td>10.5</td>
</tr>
<tr>
<td>6: Loops</td>
<td>21.1</td>
</tr>
<tr>
<td>7: Func. &amp; Proc.</td>
<td>17.9</td>
</tr>
<tr>
<td>8: Text Files Data Types</td>
<td>26.9</td>
</tr>
<tr>
<td>9: Arrays</td>
<td>31.0</td>
</tr>
<tr>
<td>10: Multidim. Arrays</td>
<td>13.3</td>
</tr>
<tr>
<td>11: Records</td>
<td>31.6</td>
</tr>
<tr>
<td>12: Files</td>
<td>16.0</td>
</tr>
<tr>
<td>13: Sets</td>
<td>4.8</td>
</tr>
<tr>
<td>14: Dynamic Struc.</td>
<td>59.1</td>
</tr>
</tbody>
</table>

CLDs brought the picture into sharper focus (Table 5). The topic of dynamic structures (Chapter 14) touches many higher level concepts. The heavy concentration of items in
Table 6

Percentage CLD Distribution by Chapter: Koffman (CS I)

<table>
<thead>
<tr>
<th>Chapter Topic</th>
<th>Kn</th>
<th>Co</th>
<th>Ap</th>
<th>An</th>
<th>Sy</th>
<th>Ev</th>
<th>n</th>
</tr>
</thead>
<tbody>
<tr>
<td>1: General Intro</td>
<td>64.3</td>
<td>21.4</td>
<td>0</td>
<td>0</td>
<td>14.3</td>
<td>0</td>
<td>14</td>
</tr>
<tr>
<td>2: Design, Debug</td>
<td>64.3</td>
<td>21.4</td>
<td>0</td>
<td>0</td>
<td>14.3</td>
<td>0</td>
<td>7</td>
</tr>
<tr>
<td>3: Control Stmts &amp; Procedures</td>
<td>12.5</td>
<td>0</td>
<td>37.5</td>
<td>12.5</td>
<td>37.5</td>
<td>0</td>
<td>8</td>
</tr>
<tr>
<td>4: Data Types</td>
<td>11.1</td>
<td>0</td>
<td>55.6</td>
<td>11.1</td>
<td>22.2</td>
<td>0</td>
<td>9</td>
</tr>
<tr>
<td>5: Case Stmts, Loops, Func.</td>
<td>22.2</td>
<td>22.2</td>
<td>44.4</td>
<td>11.1</td>
<td>0</td>
<td>11.1</td>
<td>9</td>
</tr>
<tr>
<td>6: Arrays</td>
<td>14.3</td>
<td>14.3</td>
<td>28.6</td>
<td>0</td>
<td>42.9</td>
<td>0</td>
<td>7</td>
</tr>
<tr>
<td>7: Records</td>
<td>33.3</td>
<td>16.7</td>
<td>50.0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>6</td>
</tr>
<tr>
<td>8: Sets &amp; Files</td>
<td>0</td>
<td>42.9</td>
<td>28.6</td>
<td>14.3</td>
<td>14.3</td>
<td>0</td>
<td>7</td>
</tr>
<tr>
<td>9: Recursion, Sort, &amp; Search</td>
<td>20.0</td>
<td>0</td>
<td>80.0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>10: Dynamic Struc.</td>
<td>20.0</td>
<td>20.0</td>
<td>0</td>
<td>40.0</td>
<td>20.0</td>
<td>0</td>
<td>5</td>
</tr>
</tbody>
</table>

The general introductory chapter would emphasize broad concepts and terminology. The high percentage of Knowledge level questions was not unexpected for that topic. However, the Knowledge category (59%) would be appropriate only if the chapter content were written at a similar level. The Koffman text was structured along broader conceptual lines than the other CS I books. The test bank reflected this design (Table 6). Single chapters encompassed topics
Table 7
Percentage CLD Distribution by Chapter: Lamb (CS I)

<table>
<thead>
<tr>
<th>Chapter Topic</th>
<th>Kn</th>
<th>Co</th>
<th>Ap</th>
<th>An</th>
<th>Sy</th>
<th>Ev</th>
<th>n</th>
</tr>
</thead>
<tbody>
<tr>
<td>2: Pascal Intro</td>
<td>0</td>
<td>50.0</td>
<td>0</td>
<td>50.0</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>4: Procedures</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100.0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>6: Proc. &amp; Func.</td>
<td>0</td>
<td>100.0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>7: Unconditional Loops</td>
<td>0</td>
<td>0</td>
<td>50.0</td>
<td>0</td>
<td>50.0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>8: Decision Stmts</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>50.0</td>
<td>50.0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>9: Conditional Loops</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>50.0</td>
<td>50.0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>10: Input/Output</td>
<td>0</td>
<td>0</td>
<td>100.0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>11: Case Stmts</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100.0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>13: Arrays</td>
<td>0</td>
<td>100.0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>14: Multidim. Arrays</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100.0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>15: Records</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100.0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

treated separately in other texts. This may have partially accounted for the broad ranges of CLD frequencies (9.9% for lower levels and 16.6 for higher levels). Noteworthy in this test bank was the presence of many higher level items for topics typically requiring greater conceptualization, planning, and judgment. This included discussions of
arrays, records, sets, files, recursion, sorting, searching, and dynamic structures in chapters 6 through 10.

The entire set of available questions for the Lamb test bank was assessed by the rating panel (Table 7). Because of the low number of questions provided over 15 chapters, little substantive comment could be made about the findings. All 17 items were in the taxonomic range from Comprehension through Synthesis. Matching specific course objectives and the CLD distribution of these questions would not be a practical exercise in the textbook adoption process. Using this book, an instructor would virtually be forced to construct examinations from scratch.

Computer Science II

Item distribution across the Computer Science II test banks consistently declined through the lower three levels of the taxonomy (see Figure 2). Knowledge level questions were again the most prevalent type. Raters found test items at this level more than twice as often as Comprehension questions.

Synthesis level questions dominated the higher taxonomic categories. Over 80 percent of all questions at these levels were Synthesis items. Overall, only six percent of the items rated were Analysis or Evaluation type, slightly more items occurring in the latter group.

As shown in Figure 2 and noted above, frequencies dropped rapidly at the lower levels and were erratic at the
Figure 2. Distribution of cognitive level demands for test items of Computer Science II textbooks.

upper levels. However, the cumulative percentage of items occurring at or below Application was the same in both CS I and CS II (83%). Given the different curricular emphasis of the courses, this similarity seemed inappropriate.

The CS I curriculum focuses on the Pascal syntax and grammar, programming style, and programming fundamentals.
Mastery of the basics of Pascal programming is an implicit prerequisite of CS II. Although the courses cover many of the same topics, more substantive coverage would be expected in CS II. Content is extended to include study of data structures, data management, dynamic variables, and the development of algorithms employing these topics. These tasks generally demand higher level thinking.

The preponderance of Synthesis questions at the upper taxonomic levels provided some evidence of this theoretical difference. However, the increase in Knowledge items was contradictory. The picture for CS II may be an indication both of the general difficulty of writing higher level
questions and of a repetition of CS I skills in two of the three textbooks. The test banks differed sharply from one another in CLD distribution.

The distribution of CLDs by textbook is summarized in Table 8. The vast majority of questions in the Graham text were rated at the Knowledge level (72%). Comprehension items absorbed an additional 20 percent. The majority of the remaining seven percent were in the Evaluation category. Because the Essential Elements for CS II suggest an emphasis toward higher cognitive processes, the accumulation of items at the low end was unexpected.

The test bank for Naps and Singh was also heavily concentrated at the lower taxonomic levels, although not so markedly as in the Graham text. Slightly more than half the rated test items (53.3%) were Knowledge. The three
lower levels covered 87 percent of the questions. Higher level items were focused on the Synthesis category.

Balance of lower taxonomic levels was achieved best in the Peters text. Frequencies ranged from a low of 22.6 percent (Application) to 26.2 percent (Knowledge). Higher level items were heavily concentrated in the Synthesis category (19.0%).

Again using the ranges of CLD frequencies at lower and upper cognitive levels as an indication, the differences between test banks stood out (Table 9). The heavy lower level concentration of Graham evaluation items was accompanied by a very wide range (72.4%), nearly twice that of the Nape and Singh text (37.3%) and 20 times greater than Peters (3.6). As a consequence, the range in the higher categories was very low for Graham (5.2%). The presence of a high frequency of Synthesis items caused the Peters text to exhibit the broadest high-level range (16.6%). Overall, however, the Peters test bank had a range of only 23.8 percent.

Considered by chapter, the CS II test banks provided a mixed picture of CLD distribution. The Essential Elements stressed development of programming skills and knowledge involving dynamic and static data structures, algorithm development and implementation, and the use of advanced design techniques. Viewed against this backdrop, the increase in Knowledge level test items over CS I test banks
Table 10

Percentage CLD Distribution by Chapter: Graham (CS II)

<table>
<thead>
<tr>
<th>Chapter Topic</th>
<th>Kn</th>
<th>Co</th>
<th>Ap</th>
<th>An</th>
<th>Sy</th>
<th>Ev</th>
<th>n</th>
</tr>
</thead>
<tbody>
<tr>
<td>1: Intro</td>
<td>100.0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>2: Algorithms, Recursion</td>
<td>33.3</td>
<td>67.7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>3: Hardware</td>
<td>57.1</td>
<td>42.9</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>7</td>
</tr>
<tr>
<td>4: Information Representation</td>
<td>100.0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>5: Software</td>
<td>80.0</td>
<td>20.0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>6: Values, Constants, &amp; Expressions</td>
<td>83.3</td>
<td>16.7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>6</td>
</tr>
<tr>
<td>7: Variables, I/O, &amp; Assignment</td>
<td>75.0</td>
<td>25.0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>8: Looping</td>
<td>80.0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20.0</td>
<td>5</td>
</tr>
<tr>
<td>9: Decisions</td>
<td>100.0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>10: Func., Proc., &amp; Recursion</td>
<td>33.3</td>
<td>33.3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>33.3</td>
<td>6</td>
</tr>
<tr>
<td>11: Design, Testing</td>
<td>100.0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>7</td>
</tr>
<tr>
<td>12: Types, Files, &amp; Records</td>
<td>25.0</td>
<td>50.0</td>
<td>0</td>
<td>25.0</td>
<td>0</td>
<td>0</td>
<td>4</td>
</tr>
</tbody>
</table>

remained difficult to justify. The CS II textbook did backtrack on CS I topics to some extent. However, all three focused squarely on the advanced topics and techniques.
Table 11
Percentage CLD Distribution by Chapter: Naps/Singh (CS II)

<table>
<thead>
<tr>
<th>Cognitive Level</th>
<th>Kn</th>
<th>Co</th>
<th>Ap</th>
<th>An</th>
<th>Sy</th>
<th>Ev</th>
<th>n</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chapter Topic</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1: Overview</td>
<td>80.0</td>
<td>0</td>
<td>20.0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>2: Linked Lists</td>
<td>0</td>
<td>33.3</td>
<td>66.7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>3: Strings</td>
<td>100.0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>4: Queues &amp; Stacks</td>
<td>0</td>
<td>0</td>
<td>100.0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>5: Stack Uses</td>
<td>81.8</td>
<td>18.2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>11</td>
</tr>
<tr>
<td>6: Trees</td>
<td>33.3</td>
<td>0</td>
<td>66.7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>7: Tree Variants</td>
<td>50.0</td>
<td>25.0</td>
<td>25.0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>8: Arrays &amp; Sparse Matrices</td>
<td>22.2</td>
<td>33.3</td>
<td>0</td>
<td>0</td>
<td>33.3</td>
<td>11.1</td>
<td>9</td>
</tr>
<tr>
<td>9: Graphs &amp; Networks</td>
<td>33.3</td>
<td>33.3</td>
<td>33.3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>10: Sorting</td>
<td>50.0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>50.0</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>11: Searching</td>
<td>60.0</td>
<td>20.0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20.0</td>
<td>5</td>
</tr>
<tr>
<td>12: Data Structure Management</td>
<td>80.0</td>
<td>0</td>
<td>20.0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>5</td>
</tr>
</tbody>
</table>

Chapterwise CLD distributions were difficult to reconcile with the higher level implications of the Essential Elements.

Of all the test banks involved in this study, the Graham text most heavily utilized Knowledge level questions (Table 10). Topics most conducive to upper level
Table 12

*Percentage CLD Distribution by Chapter: Peters (CS II)*

<table>
<thead>
<tr>
<th>Chapter Topic</th>
<th>Kn</th>
<th>Co</th>
<th>Ap</th>
<th>An</th>
<th>Sy</th>
<th>Ev</th>
<th>n</th>
</tr>
</thead>
<tbody>
<tr>
<td>1: General Intro</td>
<td>20.0</td>
<td>20.0</td>
<td>10.0</td>
<td>10.0</td>
<td>40.0</td>
<td>0</td>
<td>10</td>
</tr>
<tr>
<td>2: Pascal Intro</td>
<td>30.0</td>
<td>20.0</td>
<td>30.0</td>
<td>0</td>
<td>20.0</td>
<td>0</td>
<td>10</td>
</tr>
<tr>
<td>3: Procedures</td>
<td>14.3</td>
<td>57.1</td>
<td>14.3</td>
<td>0</td>
<td>0</td>
<td>14.3</td>
<td>7</td>
</tr>
<tr>
<td>4: Control Structures</td>
<td>0</td>
<td>25.0</td>
<td>50.0</td>
<td>0</td>
<td>25.0</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>5: Scalar Data Types</td>
<td>33.3</td>
<td>33.3</td>
<td>0</td>
<td>0</td>
<td>33.3</td>
<td>0</td>
<td>6</td>
</tr>
<tr>
<td>6: Arrays</td>
<td>25.0</td>
<td>25.0</td>
<td>0</td>
<td>0</td>
<td>50.0</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>7: Functions</td>
<td>33.3</td>
<td>33.3</td>
<td>33.3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>8: Records &amp; Sets</td>
<td>20.0</td>
<td>20.0</td>
<td>40.0</td>
<td>20.0</td>
<td>0</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>9: Stacks &amp; Queues</td>
<td>25.0</td>
<td>25.0</td>
<td>0</td>
<td>25.0</td>
<td>25.0</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>10: Searching</td>
<td>0</td>
<td>0</td>
<td>40.0</td>
<td>0</td>
<td>60.0</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>11: Sorting</td>
<td>20.0</td>
<td>0</td>
<td>40.0</td>
<td>20.0</td>
<td>0</td>
<td>20.0</td>
<td>5</td>
</tr>
<tr>
<td>12: Recursion</td>
<td>60.0</td>
<td>0</td>
<td>40.0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>13: Sequential Files</td>
<td>50.0</td>
<td>50.0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>14: Linked Lists</td>
<td>0</td>
<td>40.0</td>
<td>40.0</td>
<td>0</td>
<td>20.0</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>15: Trees</td>
<td>57.1</td>
<td>28.6</td>
<td>14.3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>7</td>
</tr>
</tbody>
</table>
treatment, such as information representation (Chapter 4), algorithm development and recursion (Chapter 2), and design and testing (Chapter 11), were heavily concentrated at this level. A total absence of Analysis and Synthesis items was noted. Similarly, a very small percentage of questions (5.2%) were placed in the Evaluation class.

The test bank for Naps and Singh exhibited a slightly smoother CLD distribution (Table 11). The percentage of items classified as Knowledge level was still quite high (53.3%). In the cases of the general overview (Chapter 1) and the presentations on strings (Chapter 3) and the uses of stacks (Chapter 5), the heavy concentration at this level was understandable. However, discussions of linked lists (Chapter 2) and of the topics in chapters six through twelve, higher level processes were implied by the Essential Elements. In these cases, test treatment remained in the lower three taxonomic levels. Only three of the final five chapters tapped cognition at the Synthesis or Evaluation levels.

Immediately apparent on Table 12 was the broader spread of test item CLDs for the Peters text. The only topics for which at least 50 percent of the questions fell at Knowledge level were recursion (60%), sequential files (50%), and trees (57%). Dispersion of items across the lower taxonomic levels was roughly equivalent for the advanced topics as well as for those overlapping CS I in the first
seven chapters. Furthermore, test questions dealing with arrays, searching, sorting, and linked lists reached into higher cognitive levels. Frequent use was made of Synthesis questions.

Summary

This chapter presented the findings of the study. Distributional characteristics of test banks for Computer Science I and Computer Science II were considered separately. In each case, the overall distribution of cognitive level demands was presented first. Distributions were subsequently subdivided by test bank. Finally, the dispersion of CLDs by chapter within each bank was presented and discussed.

Heavy emphasis on Knowledge level questions was found in both courses. This had not been unexpected, particularly for CS I. However, because CS II dealt with topics of a more conceptual nature, the 10 percent increase in Knowledge items detected for those test banks was unexpected. Overall, the two courses had nearly equivalent frequencies at the lower levels (82.9% and 83.4%, respectively).

Rapid but steady decrease in frequency characterized the CS I test banks. By contrast, the CS II banks contained a heavy concentration of Synthesis questions, nearly as high a frequency as Application (10.6% and 14.3%, respectively). Only six percent of the items were assessed as Analysis or Evaluation.
CHAPTER V

SUMMARY, FINDINGS, AND RECOMMENDATIONS

The purpose of this study was to produce a description of the cognitive level demands of test items in state-adopted computer science textbooks. The research question addressed the CLD distribution characteristics of the test banks accompanying those books.

Summary of the Study

This study focused solely on the test banks of textbooks adopted for use in Texas public schools. The data for this study were taxonomic levels assigned independently by a group of raters to 998 test items. Fifty percent of the questions for each chapter were randomly selected.

A pool of 13 trainees was recruited from the general population of the University of North Texas. These people were faculty, staff, graduate students, and upperclass undergraduate students. Instruction and practice in the identification and assignment of taxonomic levels was provided by a specialist in educational measurement in a two-hour workshop. At the end of the workshop, each trainee evaluated fifty test items of known characteristics. The ten raters selected for participation in the study correctly identified the taxonomic levels of at least 80% of the test
items. The rating group consisted of one faculty member, six graduate students, and three undergraduates with upperclass standing.

Descriptive statistics appropriate for ordered data were computed using SPSS/PC+ (version 2.1), the microcomputer version of the Statistical Package for the Social Sciences (Norusis, 1986). Central tendency was described by medians and modes; dispersion was indicated by calculations of frequency ranges. Skewness was computed as an indicator of shape.

Findings

Although no specific theoretical distribution of taxonomic levels had been suggested for either computer science course, the general trend was expected to be hyperbolic. The majority of test items were anticipated at the lower levels. The frequency curve was envisioned as dropping rapidly through these categories, producing a broad frequency range. The higher cognitive processes were pictured as having a flatter frequency degradation and narrower range.

The observed characteristics sustained the expectation only partly. Both courses exhibited decreases in frequency progressing through the three lower levels. However, the drop from Knowledge to Comprehension in CS II test banks was much steeper than in CS I. The percentage of questions
falling in these levels was approximately 83 percent for both courses.

At the higher levels, the decline was roughly five percent per level in CS I. However, in CS II there was no clear trend. Over 10 percent of the test items were assessed as Synthesis level. This was more than three times the frequency of either Evaluation or Analysis. This condition is intuitively sound. Problem solving through programming is an integral part of the CS II curriculum. The process of creating new entities (in this case new programs), of using a wide range of skills to achieve a goal, would be expected more often in CS II.

The range of frequencies in CS I varied markedly among the four test banks surveyed. The frequency range at lower taxonomic levels was greatest with the Mandell text (50.9%) and least with the Nance text (3.0%). At the upper levels, the Mandell text had the lowest range (9.4%); the widest was in the Lamb text (29.4%). The low value of the Mandell bank was undoubtedly due to the high concentration of items at the lower levels. Overall, the Koffman text had the narrowest range (21.8%).

Similarly, the frequency ranges for CS II exhibited high variability. The Graham text had a frequency range in excess of 72 percent for the lower taxonomic levels. Naturally, the upper level range was quite small (5.2%). The narrowest low-level range was found with the Peters book
(3.6%), which also had the greatest high-level range (14.2%).

The implications of these findings are subtle. Bloom’s taxonomy was intended for use by curriculum designers. By selecting activities at various cognitive levels, students could be led from simple knowledge acquisition to concept mastery. Course objectives would define the course. Testing instruments would gauge progress by examining achievement of the various objectives. The results of this study offer little encouragement for the use of publisher-supplied test banks.

Objectives matching the reviewed test banks would place heavy emphasis on lower level cognitive skills. Computer science is more than programming. The Texas CS I and CS II curricula recognize this fact. Computer instruction, especially beyond a course in computer awareness, has questionable value if not built around problem solving (Upchurch & Lochhead, 1987). The thrust of the Texas computer science courses is toward problem solving. Low level objectives cannot achieve these ends (Pea & Kurland, 1983). A course built to match the test banks of the state-adopted texts would also fall short of this goal.

This study has laid the foundation for using test bank analysis in the textbook adoption process. At the very least, the State Board of Education could make distribution characteristics known to local school boards. The analysis
process is not complicated; no target distribution need be specified. But district selection committees would have one more tool at their disposal when evaluating texts.

The seven test banks have not been labelled as "good" or "bad." None was termed "better than" any other. Such qualitative judgments were beyond the scope and intent of this study. Furthermore, what may appear objectionable on the surface may be perfectly acceptable to some school district.

However, there is one other subtle factor in the use of the test banks. The Texas Education Agency has established appropriate certifications for both computer courses. However, a recent study of the 1,305 secondary schools in Texas found only 59 percent of CS I teachers and 57.5 percent of CS II teachers appropriately certified (Brown, 1988). Computer science is a complex field of study. Rather than relying on limited knowledge of the nuances of the various topics for creating tests, uncertified teachers may perceive the supplementary test banks as more valid. If this reliance on publisher-supplied questions is high, the results of this study certainly underscore the need for employing certified teachers. With more comprehensive preparation in both computer science and test preparation, some of the more glaring disparities in taxonomic distributions may be avoided.
What has been accomplished is the germinal work to establish an evaluation system for computer science test banks. Carried to completion, the process could be used in the adoption cycle beginning in 1991. Before that can happen, however, additional research in several areas is indicated.

Recommendations for Further Study

Informal interviews were held with the raters after all data had been collected. Each was invited to comment freely about the study. Three topics recurred often enough to warrant additional study: subject-specific key words, computer jargon, and the workshop content and format.

In the workshop, raters had been taught to use certain key words as probable indicators of the various taxonomic levels. For instance, Analysis level was associated with such words as "associate," "determine," "differentiate," and "outline." In practice, however, raters said they often found the keys to be phrases or types of directives. For example, each of the following was associated with an Analysis level rating in the study:

--- "The X field of the second element of the array NUMARY is referred to by . . ."
--- "Evaluate the following boolean expressions."
--- "What is the purpose of this procedure?"

None of the anticipated key words for Analysis level appeared in these items. Nonetheless, the rater in each
case perceived a reasoning function being evoked. Further research is suggested for the purpose of associating the levels of Bloom's taxonomy with key words and phrases typical of computer science and programming examinations.

The subject of computer jargon is directly related to raters' computer awareness. Raters reported a wide variation in computer expertise. At the low end were four individuals who were reasonably comfortable with computer use but had little or no programming experience or formal training. The remaining raters were experienced programmers. These six felt their familiarity with computer jargon helped them extract the intent of test items. The four less experienced raters were frequently unsure of questions. Therefore, studies should be conducted to determine whether interrater reliability is affected by programming experience.

Future researchers may want to modify the taxonomy workshop. Participants found the session informative, interesting, and potentially useful. Even people not selected as raters were intrigued by the entire concept. However, future workshops need to focus more directly on computer key words.

The results of this study offer little encouragement for the use of publisher-supplied test banks. However, the reported information could be put to effective use by authors, publishers, and textbook evaluators. The State
Board of Education could make distribution characteristics known to local school boards. With further research as a basis, publishers and test authors could design test banks with broadly based taxonomic demands to the ultimate benefit of students and children.
APPENDIX A

Computer Science Textbooks Adopted in Texas

Computer Science I


Computer Science II


64

Texas Essential Elements for Computer Science

75.124 Computer Science.

(a) Computer Science I (1 unit). Computer Science I shall include the following essential elements.

(1) Beginning concepts associated with programming methodology. The student shall be provided opportunities to:

(A) develop functional specifications for specific programming problems;

(B) use current program design methodology [sic] such as modularization, top-down design, and stepwise refinement to develop program solutions for a given problem specification;

(C) develop structured program coding with good style and clarity of expression;

(D) demonstrate skill in testing for program correctness, using effective coding, design, and test data;

(E) develop effective debugging strategies; and

(F) develop adequate internal and external documentation.
(2) Beginning concepts and skills associated with programming languages. The student shall be provided opportunities to:

(A) code with a block-structured language using both local and global identifiers correctly;

(B) develop coding with correct and efficient use of data as represented by constants and variables;

(C) develop coding with correct and efficient use of expressions and assignment statements, including the use of standard functions, operators, and proper operator precedence;

(D) develop coding with correct use of sequential, conditional, and repetitive execution control structures;

(E) demonstrate effective use of predefined input and output procedures for a language;

(F) develop coding with effective use of procedures and functions, identifying actual and formal parameters and using properly value and reference parameters; and

(G) annotate coding properly with comments, indentation, and formatting.
(3) Beginning concepts and skills associated with data
types and structures. The student shall be
provided opportunities to:
(A) develop coding using the primitive data types
for numeric data, character data, and Boolean
data; and
(B) develop coding using the linear data
structures: arrays and strings.

(4) Beginning concepts and skills associated with
algorithms. The student shall be provided
opportunities to:
(A) solve programming problems using sequential
and iterative algorithms; and
(B) design and code sequential (linear) search
algorithms to be used in solving problems for
information storage and retrieval.

(5) Beginning concepts associated with the
applications of computing. The student shall be
provided opportunities to:
(A) code and study examples of coding for text
processing;
(B) code or study examples of coding for
simulation and modeling;
(C) code and study examples of coding for data
analysis;
(D) code and study examples of coding for data management;

(E) code or study examples of coding for system software and graphics; and

(F) code or study examples of coding for games.

(6) Beginning concepts associated with computer systems. The student shall be provided opportunities to:

(A) identify major hardware components such as processors, peripherals, and memory; and

(B) identify and use system software such as language processors, operating systems, and graphical output facilities.

(7) Beginning concepts associated with the social implications of computers. The student shall be provided opportunities to:

(A) discuss the responsible use of computer systems; and

(B) discuss the social ramifications of computer applications related to privacy, values, and reliability of systems.

(b) Computer Science II (1 unit). Computer Science II shall include the following essential elements.

(1) Concepts associated with programming methodology [sic]. The student shall be provided opportunities to:
(A) develop larger programs with increased emphasis on design, style, clarity of expression, and documentation as they relate to ease of maintenance, program expansion, reliability, and validity;

(B) develop further skill in testing programs, including methodologies for program verification; and

(C) perform analysis of design methodologies such as top-down design versus bottom-up.

(2) Concepts and skills associated with programming languages. The student shall be provided opportunities to:

(A) develop coding flexibility with file input and output;

(B) compare various programming languages and analyze their appropriateness for a variety of applications; and

(C) develop coding skills with recursive procedures and algorithms.

(3) Concepts and skills associated with data types and structures. The student shall be provided opportunities to:

(A) develop coding using the linear data structures: arrays, strings, linked lists, stacks, and queues;
(B) develop coding using binary tree structures and effectively use current terminology related to tree data structures; and

(C) develop skills in representing data structures sequentially, with pointers, and with linked data structures.

(4) Concepts and skills associated with algorithms [sic]. The student shall be provided opportunities to:

(A) design and code programming solutions requiring the use of binary search, hash-coded search, and the search of an ordered binary tree;

(B) analyze the benefits and applicability of the various search algorithms with respect to search time;

(C) design and code programming algorithms using bubble sort, merge sort, sorting using an ordered binary tree, and quicksort; and

(D) develop skills in the manipulation of data structures including string processing, insertion and deletion in linear structures and trees, and tree traversals.

(5) Concepts associated with numerical algorithms. The student shall be provided further opportunities to:
(A) code or study code for programming solutions by approximation;

(B) code or study code for programming solutions using statistical algorithms; and

(C) discuss the importance of the numerical accuracy of computers in designing algorithms.

(6) Concepts associated with the applications of computing. The student shall be provided further opportunities to:

(A) code programming solutions for text processing;

(B) code programming solutions for simulation and modeling;

(C) code programming solutions for data analysis;

(D) code programming solutions for data management;

(E) code programming solutions for system software and graphics; and

(F) code programming solutions for games.

(7) Concepts associated with computer systems. The student shall be provided opportunities to:

(A) use and be able to explain the function of interpreters and compilers which make it possible for them to execute the programs they write; and
(B) analyze the trade-offs made in microprocessor-based personal computers and in larger multi-user systems among factors such as cost, storage capacity, execution speed, remote access, and the ability to share files.

(8) Concepts associated with the social implications of computers. The student shall be provided opportunities to:

(A) discuss the responsible use of computer systems; and

(B) discuss the social ramifications of computer applications related to privacy, values and reliability of systems.

Source: *State Board of Education Rules for Curriculum* (Texas Education Agency, 1985)
REFERENCES


State Board of Education (1985). *Proclamation of the state board of education advertising for bids on textbooks, No. 62.* (Available from the Texas Education Agency, Austin, TX)


Texas administrative code, Title 19, Chapter 81, Subchapter D, State textbook program (1985).


