DO COMPUTER SCIENCE. INFORMATION SYSTEMS. AND SOFTWARE ENGINEERING PROFESSIONALS ACCEPT A COMMON BODY OF COMPUTING KNOWLEDGE?

A Thesis

Submitted to the Graduate Faculty of the
University of South Alabama
in partial fulfillment of the
requirements for the degree of

Master of Science

in

The School of Computer and Information Sciences

by
Ronnie L. Williams
B.S.. University of South Alabama. 1970
December. 1997

UMI Number: 1387221

THE UNIVERSITY OF SOUTH ALABAMA

COLLEGE OF ARTS AND SCIENCES

DO COMPUTER SCIENCE, INFORMATION SYSTEMS, AND SOFTWARE ENGINEERING PROFESSIONALS ACCEPT A COMMON BODY OF COMPUTING KNOWLEDGE?

BY

Ronnie L. Williams

A Thesis

Submitted to the Graduate Faculty of the
University of South Alabama
in partial fulfillment of the
requirements for the degree of

Master of Science

in

The School of Computer and Information Sciences

December, 1997

Approved: Date:

Chair of Thesis Committee

Committee Member

Committee Member

Committee Member

Committee Member

Department Chair

Director of Graduate Studies

Dean of the Graduate School 11-4-97

To Houriya

ACKNOWLEDGMENTS

I want to express my appreciation to the members of my thesis committee for their assistance with this thesis. I especially thank my committee chairman, Dr. Longenecker, without whose assistance this thesis would not have been possible.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Williams, Ronnie Lorell, M.S., University of South Alabama, December, 1997. Do Computer Science, Information Systems, and Software Engineering Professionals Accept a Common Body of Computing Knowledge? Chair of Committee: Herbert E. Longenecker, Jr.

Longenecker and Williams (Longenecker et al 1995) generated a common body of computing knowledge from curriculum documents for Computer Science, Information Systems, and Software Engineering. An abstraction of this body of knowledge was developed and used to survey approximately 1,000 computing professionals for their expectations for knowledge elements in the three computing programs for students graduating from four-year undergraduate universities. It is the general hypothesis of this thesis that results of the survey reveal a core of similar expectations as well as important differences for the three programs, and that computing professionals accept a common body of computing knowledge.

1. INTRODUCTION

Computer science is an accredited computing program of the Computer Science Accreditation Commission of the Computing Services Accreditation Board (CSAC/CSAB), with 129 institutions accredited to teach the program in the United States, as of May, 1996 (LaMalva and Peterson 1996). Information systems and Software Engineering are not separately accredited computing programs, but separate curricula have been extensively developed for each since the 1970s.

A problem exists because no agreed upon areas of difference or similarity have been identified for computing curricula for the three programs, and the names "computer science," "information systems," and "software engineering" are sometimes used almost interchangeably. The problem with content of computing programs is evidenced by employment advertisements for computing professionals placed in major newspapers (Washington Post 1997), many or most of which could be filled by persons with degrees in computer science, information systems, or software engineering, interchangeably. In these advertisements, jobs with very similar computing knowledge requirements are frequently listed side-by-side under different program names (See Figure 1.).

The above problem could be initially addressed by surveying computing professionals in academia to determine how they define the curricula requirements for the three computing programs. This method assumes that each of these professionals would be able to identify themselves with one of the three programs, and that they would be able to identify curriculum requirements for the programs based on their respective teaching experience or expertise.

One way to facilitate this process would be to develop a single or combined "body of knowledge" which contains most of the individual knowledge elements which could possibly comprise the curricula for the three separate programs. Computing professionals could then be asked to identify which of these elements fall within the areas in which they teach or have expertise, and the degree of learning required for students for each. Similarities and differences in the three programs could then be determined.

In 1995 a combined body of knowledge was prepared using curriculum documents previously developed for the three computing programs. This combined body of knowledge was then abstracted and used in a "pilot" survey of approximately 1,000 computing professionals (Longenecker et al 1995). The purpose of this survey was to develop an initial overall perspective of the similarities and differences in the bodies of knowledge for computer science, information systems, and software engineering. Using the results of this survey, future research efforts can be undertaken to determine these similarities and differences more precisely.

This thesis analyzes the results of the pilot survey, presents them for use in future research efforts, and offers an initial answer to the question, "Do computer science, information systems, and software engineering professionals accept a common body of computing knowledge?"

POSITION TITLE	
Analytic Database Coordinator	Software Developer
Application Developer	Software Development Engineer
Application Software Engineer	Software Engineer (Application Developer)
Business Data Analyst	Software Engineer (Information Engineer)
Business Process Re-Engineer	Software Engineer (Programmer Analyst)
Database Analyst/Developer	Software Engineer (Systems Programmer)
Database Administrator	Software Implementation Consultant
Firmware Engineer	Software Quality Engineer
Information Engineer	Systems Engineer (Data Engineering)
Mainframe Developer	Systems Engineer (Network Architecture)
Network Engineer	System Administrator
Network Management System Software Engineer	System Analyst
Process Control Engineer	System Architect
Programmer/Analyst	Systems Integrator
Software Architect	Technical Support Engineer

The above list is a small sample of computing position titles contained in the employment section of The Washington Post, dated September 7, 1997. That employment section contained 41 pages devoted entirely to computing jobs.

Few positions were listed under computer science, information systems, or software engineering, per se, and most did not specify a requirement for a particular degree or even any degree or university training. Most did have requirements for specific systems or programming language experience or for professional certifications and technical computing experience. Qualifications for developers, analysts, programmers, engineers, administrators and others were often every similar, even within the same advertisement (for multiple positions), with no indication of why the position titles were different. The four very different titles listed for software engineer were contained in a single advertisement.

Most of the positions listed could be filled with persons identifying themselves as computer scientists, information systems professionals, or software engineers. The most often used degree requirement for positions requiring a degree was, "Bachelor of Science in Computer Science or a related degree."

Figure 1. List of computing position titles

2. BACKGROUND

Significant curriculum development efforts began in 1968 for computer science, in 1972 for information systems, and in 1976 for software engineering. Following is a summary of these and other events and efforts, which contribute to this thesis:

2.1. Computer Science

Development of a curriculum for computer science (See Figure 2.) began in 1968, when the Association for Computing Machinery (ACM) published "Curriculum 68: Recommendations for the undergraduate program in computer science" (ACM 1968). This curriculum was revised in 1978 (ACM 1979), and then again in 1991 by a joint task force of the ACM and the Computer Society of the Institute of Electrical and Electronics Engineers (Turner and Tucker 1991). The 1991 revision includes a listing of common "requirements" and "knowledge units" which should be addressed during development of undergraduate curricula in computer science. These requirements and knowledge units are incorporated into the computer science content of the bodies of knowledge referenced by this paper (Davis et al 1997).

2.2. Information Systems

A curriculum for information systems was first developed in the early 1970s (Ashenhurst 1972; Couger 1973). These efforts (See Figure 2.) were followed by publication of curricula models in the 1980s by both the Data Processing Management Association (DPMA) and the ACM (DPMA 1981, 1986; Nunamaker, Couger and Davis 1982). Beginning in 1988, DPMA supported a Curriculum Task Force (IS'90) to revise the 1986 curriculum. It included members from the ACM and the Institute for Certification of Computer Professionals (ICCP). This group's efforts resulted in the publication of "IS'90:

The DPMA Model Curriculum for Information Systems for 4 Year Undergraduates" (Longenecker and Feinstein 1991). In 1993, another task force, consisting of members from the Association of Information Technology Professionals (AITP) (formerly DPMA), ACM, and the Association for Information Systems (AIS) published a comprehensive four-year curriculum model for information systems, building on all previous models (Davis et al 1997). (The IS97 task force continues work published as IS'95, and then as IS'96. References in this thesis to IS'95 and IS'96 refer to on-going efforts contributing to evolution of IS'97.)

2.3. Software Engineering

In 1976, work began in earnest on curricula development for the new discipline of software engineering (See Figure 2.) (in Glass 1992; Freeman, Wasserman, and Fairley 1976), but it was not until 1978 that the first published curriculum proposal was developed (in Glass 1992; Freeman and Wasserman 1978), later to became an unpublished draft report (in Glass 1992; IEEE 1980). It was only a little later, in 1984, that the United States Department of Defense funded the Software Engineering Institute (SEI) at Carnegie Mellon University. Since then, the SEI has taken the lead in software engineering curriculum development in this country, having published a number of curriculum documents, including curriculum models for both graduate and undergraduate software engineering education (Ford 1990; Ford 1991). Additionally, at about the same time, the British Computer Society (BCS) and the Institution of Electrical Engineers (IEEE) published "A report on Undergraduate Curricula for Software Engineering," which expands on earlier work and provides extensive software engineering curricula development data (BCS 1989).

2. 4. Common Body of Knowledge for Computer Science and Information Systems

Efforts began in 1994 to develop a common body of knowledge for computer science and information systems, using existing curricula documents (See Figure 3.). A graduate class at the University of South

CS	1968 - Curriculum 68: Recommendations for the Undergraduate Program in Computer Science (ACM 1968)
	1978 - 1968 curriculum revised (ACM 1979)
	1991 - 1978 curriculum revised (Turner and Tucker 1991)
IS	1972 - Curriculum Recommendations For Graduate Professional Programs in Information Systems (Ashenhurst 1972)
	1973 - Curriculum Recommendations for Undergraduate Programs in Information Systems (Cougar 1973)
	1981 - DPMA Model Curriculum, 1981 (DPMA 1981)
	1982 - Information Systems Curriculum Recommendations for the 80s: Undergraduate and Graduate Programs (Nunamaker, Couger and Davis 1982)
	1986 - DPMA Model Curriculum, 1986 (DPMA 1986)
	1991 - IS'90 The DPMA Model Curriculum for Information Systems for 4 Year Undergraduates (Longenecker and Feinstein 1991)
	1995 - Draft IS'95 Model Curriculum and Guidelines for Undergraduate Degree Programs in Information Systems (Longenecker, Feinstein, Gorgone, Davis and Couger 1995)
	1996 - Draft IS'96 Model Curriculum and Guidelines for Undergraduate Degree Programs in Information Systems (Longenecker, Feinstein, Gorgone, Davis and Couger 1996)
	1997 - IS'97 Model Curriculum and Guidelines for Undergraduate Degree Programs in Information Systems (Davis, Gorgone, Couger, Feinstein, and Longenecker 1997)
SE	1976 - Essential Elements of Software Engineering Education, Proceedings of the Second International Conference on Software Engineering (Freeman, Wasserman, and Fairley 1976)
	1978 - A Proposed Curriculum for Software Engineering Education, Proceedings of the Third International Conference on Software Engineering (Freeman and Wasserman 1978)
	1989 - Report on Undergraduate Curricula for Software Engineering, British Computer Society and The Institution of Electrical Engineers (BCS 1989)

Figure 2. Chronology of major curriculum development events

DPMA	1988	IS'90 Curriculum Task Force established.
IS'90 Task Force	1989	Developed information systems body of knowledge (IS'90).
ACM and IEEE	1991	Developed revised computer science body of knowledge (CS'91).
University of South Alabama class project, in coordination with IS'95 Task Force (coordinated by Longenecker)	1994	Matched IS'90 body of knowledge with CS'91 (IS'90/CS'91 common body of knowledge).
DPMA, ACM, and AIS	1993	Joint ACM/AIS/DPMA Curriculum Task Force established.
Longenecker and Williams	1995	Extensive clean-up of IS'90/CS'91 common body of knowledge.
Couger and team (Chicago)	1993	Minor revisions of IS'90 common body of IS knowledge.
IS'95 Task Force	1994	Extensive revisions of IS'90 common body of knowledge. Task Force adopts revised document for curriculum planning (IS'95).
Longenecker and Williams	1995	Merged CS'91 with revised IS body of knowledge to form IS'95 body of knowledge.
Longenecker	1996	Added ethics content to body of knowledge (based on National Science Foundation task force recommendation)

Note: Herbert E. Longenecker (University of South Alabama) and J. Daniel Couger (University of Colorado) served with others as Co-Chairs of the IS'95/IS'96 Curriculum Task Force. Ronnie Williams (University of South Alabama) was a member of the IS'95/IS'96 Curriculum Task Force and is author of this thesis. Couger's team worked both with IS'95/ IS'96, and independently in Chicago, to revise the body of knowledge.

Figure 3. Steps in synthesis of the common body of computing knowledge for CS and IS

Alabama analyzed the revised computer science curriculum developed by the ACM and IEEE in 1991 (Turner and Tucker 1991), in coordination with the IS'95 Curriculum Task Force. The class extracted knowledge elements from the computer science curriculum at similar levels and types of detail as contained in the most recently available information systems curriculum for four year undergraduate university programs (Longenecker and Feinstein 1991), and integrated these elements into the existing information systems body of knowledge structure. That structure was subsequently extensively reworked by different groups working under leadership of co-chairs of the IS'95 task force (Longenecker and Couger), resulting in a common body of knowledge for computer science and information systems.

2.5. Body of Knowledge for Software Engineering

As part of the preliminary work for development of its pilot survey, the IS'95 task force synthesized a body of knowledge for software engineering, in the same overall format as the common body of knowledge for computer science and information systems (mentioned above). This synthesis was explicitly derived from analysis of curriculum content contained in reports on software engineering education developed by the Software Engineering Institute (Ford 1990, 1991), assisted by the observations of Glass 1992, other reports from the SEI (Berry 1992; Ford, Gibbs, and Tomayko 1987; Ford and Ardis 1989; Ford 1994; Gibbs and Ford 1986; Shaw 1986; Shaw 1990; SEI 1991; Tomayko and Shaw 1991), and other efforts (BCS 1989; Ford and Gibbs 1989; Freeman, Wasserman, and Fairley 1976; Freeman and Wasserman 1978; Freeman 1987; Gibbs 1989; IEEE 1980; Leventhal and Mynatt 1987; NSF 1993; Parnas 1990; and Wasserman 1976).

2.6. Combined Body of Computing Knowledge for Computer Science, Information Systems, and Software Engineering

The IS'95 task force developed a combined body of knowledge which is a synthesis of the bodies of knowledge mentioned above. The combined common body of computing knowledge was included in its entirety as a part of the IS'95 pilot survey of computing persons in academia (discussed below), and is provided with this thesis as Appendix A.

The combined body of knowledge contains "knowledge elements" identified by the IS'95 task force as necessary for development of curricula for the topic areas surveyed. It contains more than 500 knowledge elements at four levels of detail, and under three major groupings (1.0. Information Technology, 2.0 Organizational and Management Concepts, and 3.0 Theory and Development of Systems). A sample of this four-level organization is shown in Figure 4.

Body of Computing Knowledge Elements in IS'95
1.0 Information technology **(Level 1)**
1.1 Computer architectures **(Level 2)**
1.1.1 Fundamental data representation: non-numeric... **(Level 3)**
1.1.1.1 Basic machine representation of numeric... **(Level 4)**

Figure 4. Example of 4-level design of the common body of computing knowledge

The major purpose of the Body of Knowledge is to provide a means for curriculum planners to identify all of the "pieces" necessary for a curriculum in any of the three topic areas. Once all of the pieces of a total curriculum have been identified, "exit levels" can be determined for each, specifying the level of learning necessary for each element, for each topic area. It is anticipated that the body of knowledge contains many elements common to all three topic areas, with each having different exit levels depending on the program area.

2.7. Summary Body of Knowledge (Used in the IS'95 Survey)

Because the body of knowledge contains so many separate elements and levels of detail, it could not easily be used in a survey. For this reason, it was condensed into 68 single-level knowledge elements (See Figure 5.). These 68 elements have a one-to-many relationship with the total body of knowledge, and can be referenced back to it without difficulty. A copy of the complete list of 68 summary elements is provided with this thesis as Appendix B. (In Figure 5, Computer Architectures is an element at level 2 of the body of knowledge. On the body of knowledge, it contains six third-level and thirty-four 4th-level elements under it. These forty elements were combined into only four summary topics. Figure 5 is an illustration of how one third-level element and five fourth-level elements were combined into one summary topic (or question) for use in the survey. A complete list is provided of the third- and fourth-level content of only one summary topic -- Summary Topic 1.

2.8. Definitions Used in the IS'95 Survey

The IS'95 task force prepared standardized definitions which were used in the IS'95 pilot survey. The definitions are contained in Figure 6.

2.9. The IS'95 Survey

The IS'95 survey was completed in July 1995. Approximately 1,000 computing academics were surveyed. The list of participants were developed primarily from membership and attendee lists of national computing organizations.

2.10. IS'95 survey Metrics

Participants were asked to respond to two types of question: (1) identifying information, such as name, computing program of primary interest, membership in organizations, etc., and (2) anticipated knowledge exit levels for summarized body of knowledge elements. The survey used modified Bloom knowledge "exit levels" as the basis for determining exit levels for summarized body of knowledge elements (Bloom 1956). Participants were asked to provide the exit levels they believed students in their area of interest should reach after four years of study. Exit levels were based on the taxonomy of educational objectives, shown in Figure 7. This modified taxonomy was used in order to be consistent with other computing surveys and reports (e.g., Longenecker and Feinstein (eds) 1991).

For each knowledge summary element listed in the survey, participants were offered five exit level choices (A through E), one of which they entered on a standard scan sheet for each listed element. One of the choices ("E") was "No Answer," giving participants an opportunity to reject a particular knowledge element as part of the body of knowledge.

The significance of the IS'95 pilot survey is that, for the first time, comprehensive body of knowledge data is available for analysis for all three of the topics: CS, IS, and SE. Analysis of this data will contribute to an understanding of the curriculum content relationships between the three topics, and it will contribute to improved future surveys. It will also contribute to answering the question, "Do computer science, information systems, and software engineering professionals accept a common body of computing knowledge?"

1.1 Computer Architectures: (Contains six third-level and thirty-four fourth-level knowledge elements. These were combined into four summary topics.)

Summary Topic 1. Fundamental data presentation and physical representation of digitized information -- numeric, non-numeric (integers, reals, errors, precision); data, text, image, voice, video

(Summary Topic 1 contains one third-level and five fourth-level elements from the body of computing knowledge, presented here in their entirety.)

1.1.1 Fundamental data representation: non-numeric, numeric (integers, reals, errors, precision)

1.1.1.1 Basic machine representation of numeric data
1.1.1.2 Basic machine representation of non-numeric data
1.1.1.3 Finite precision of integer and floating point number representation
1.1.1.4 Errors in computer arithmetic and related portability issues
1.1.1.5 Basic concepts of computer architecture

(Detailed descriptions and content is not presented for Summary Topics 2, 3, and 4.)

Summary Topic 2. CPU architectures and computer system components...

Summary Topic 3. Multiprocessor architectures -- single multiprocessing and...

Summary Topic 4. Digital logic and systems -- logic elements and switching...

Figure 5. Example of summary topics used in the survey of computing professionals

DISCIPLINE	EMPHASIS	DEFINITION
Computer Science	Information Technology and algorithm development (Not organization and management or development of systems, per se)	"the systematic study of algorithmic processes that describe and transform information: their theory, analysis, design, efficiency, implementation, and application The fundamental question underlying all of computing is "What can be (efficiently) automated?" (Denning in IS'95)
Information Systems	Application of technology to organizations and use of and development of systems (Not technology, per se)	"... complex socio-technical entities that have taken on critical roles in local, national and global organizations. Information systems provide support for the goals of the organization and its management -- strategic, tactical and operational -- in a timely and cost effective manner. Thus, the goal of these systems is "to improve the performance of people through the use of information technology...where the ultimate objective is performance improvement...where the focus is the people who makeup the organization..." (IS'95).
Software Engineering	Use of technology and development of software (Not technology or development of organizational systems, per se)	"The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software." (IEEE 1990). Ford added the following comment... The two concerns that pervade software engineering are the complex requirements of systems and the need to build them economically in a for-profit environment. The context of software engineering tends to be software intensive systems that have substantial performance (real-time), capacity, reliability, security, and safety requirements; the discipline addresses how such systems are built and maintained in ways that are economically viable for the producers and users. (Ford in IS'95)
"Emphasis" for the three definition groups was obtained as follows: (1) Emphasis for computer science and information systems was abstracted from the IS'97 Model Curriculum and Guidance for Undergraduate Degree Programs in Information Programs (Davis et al 1997); (2) Emphasis for software engineering was abstracted during synthesis of the Software Engineering Body of Knowledge (See Section 2.5.). References to Denning and Ford in IS'95 pertain to the IS'95 survey and the draft report (Longenecker et al 1995).		

Figure 6. Definitions of computer science, information systems, and software engineering

EXIT LEVELS		
SURVEY LEVEL	**BLOOM LEVEL**	**DESCRIPTION OF EXIT LEVELS**
A. Awareness	1. Knowledge Recognition	Awareness; introductory recall and recognition; recognition knowledge (Can be identified by such terms as: Define..., List characteristics of..., Name components of..., Diagram..., List advantages and disadvantages of..., Classify...).
B. Literacy	1. Differentiation	Literacy; knowledge of framework and content; differentiation knowledge (Can be identified by such terms as: Compare and contrast..., Execute simple..., Write simple..., Functional capabilities are..., Describe interrelation of object to other objects in the same context).
C. Ability to Use	2. Comprehension, Translation / Extrapolation, Use of Knowledge	Concept; comprehension and user knowledge as exemplified by translation, exploration and interpretation of meaning (Can be identified by such terms as: Communicate idea or abstraction of..., Given a ... translate it into..., Given a set of ... interpret..., Given a set of... extrapolate to..., List concepts used in..., List major steps in..., Explain..., Use/Exercise...).
D. Ability to Apply	3. Application Knowledge	Detailed understanding; appropriate application of knowledge in a structured/controlled context resulting from considerable "cultivation" (Can be identified by terms such as: Be able to write syntactically correct..., Debug..., Implement an... and maintain it, Apply principles of... to..., Design a... for...).
E. No Assumed Knowledge		The student is not expected to have any familiarity with this knowledge element.
	4. Analysis 5. Synthesis 6. Evaluation	Develop/originate/institute...; Construct/ adapt...; Generate novel solutions to...; Come up with new knowledge regarding...; Evaluate/ judge the relative value of..., with respect to...

Figure 7. Comparison of exit levels in survey versus Bloom exit levels

3. HYPOTHESES

It is the general hypothesis of this thesis that results of a survey of computer scientists, information systems professionals, and software engineers will reveal a core of similar expectations as well as important differences for the three programs and that they accept a common body of computing knowledge.

H_0 All topics of the body of computing knowledge are acceptable to all survey participants regardless of their program classification.

H_1 For persons identifying themselves with CS, it would be expected there would be an emphasis on "information technology and algorithm development" as opposed to organization and management or development of systems, per se.

H_2 For persons identifying themselves with IS, it would be expected there would be an emphasis on "application of technology to organizations" and on "use of and development of systems," rather than on technology, per se.

H_3 For persons identifying themselves with SE, it would be expected there would be an emphasis on "use of technology and development of software," rather than on technology or development of organizational systems, per se.

4. METHOD

The methodology for proving the hypotheses involves three steps: (1) processing the survey response data, (2) preparing and presenting the data for review and analysis, and (3) analyzing the data to determine if it verifies the hypotheses.

4.1. Processing Survey Response Data

Survey participants entered their responses on electronic scan sheets. These sheets were processed by a scan sheet reader, and a text file was generated for responses. The file was imported into a spreadsheet, using the Microsoft Excel® file import procedure. Using Microsoft Excel®, this spreadsheet was extended and used to develop the counts, totals, means, standard deviations, and other statistical data, as well as the tables and charts presented in the thesis.

4.2. Preparing and Presenting Survey Response Data

Response data is presented in the thesis, as follows:

4.2.1 Descriptive Statistics

A table of the 68 summary topic questions used in the survey is provided, with counts of persons responding, means, and standard deviations for each of the three computing programs for which survey data was solicited. An overall means is also provided for all responses, independent of computing program. Standard Microsoft Excel® functions such as COUNT (for total number of participants), AVERAGE (for arithmetic mean), and STDEV (for standard deviation) were used to develop this data. The table was used as a source for data used in the analyses, and it is included as Appendix C. Basic survey participation data (i.e., number of participants, overall means for questions answered, and standard deviation) was extracted and is provided as Table 1 in the Results section.

4.2.2. Graphical Plots of Means for Responses

Responses were sorted by computing program, and means were calculated for each topic within each program. One chart with no topic detail was developed to show visually the overall pattern of means for responses as they relate to each of the three computing programs, as well as to the overall means for all responses. Additionally, four detailed bar charts were developed, three separately showing the means for responses to the 68 survey questions for each of the three computing programs surveyed, and one showing the means of all responses. These charts were examined for top exit level responses (exit levels of 3 or above), and these responses were sorted in topic order and arranged in tables, in order by primary body of knowledge group (I, II, and III). Separate tables were prepared for each computing program. Charts and tables are provided in the Results section as Figures 8 and 9, and Tables 2, 3, and 4.

4.2.3. High Level Comparisons

In order to obtain a different perspective on survey responses, means data was aggregated at level-2 of the body of knowledge. Topic responses were sorted and means were calculated under each of the level-2 body of knowledge elements. These elements were then arranged in order under the three primary body of knowledge groups, and presented for each of the computing programs. One table and one chart were prepared and are provided in the Results section as Table 5 and Figure 10.

4.2.4. Major Differences ("Deltas")

Responses were sorted by computing program, and means differences (or deltas) were calculated for three comparison groups (IS - CS, IS - SE, and CS - SE). [1] One chart with no topic detail was developed

[1] Only three charts were prepared because the three other possible difference combinations (CS - IS, SE - IS, and SE - CS) are mirror images of the three charts which are provided, with a change in sign (e.g., IS - CS = "negative means" becomes CS - IS = "positive means".)

to show visually the overall pattern of differences in means for responses, as they relate to each of the three comparison groups. The chart is included as Figure 11 in the Results section

4.2.5. Sorted and Cumulative Means Comparisons

Means and cumulative means for topic responses were plotted on charts for each of the computing programs. Means were plotted in descending order of importance. Cumulative means were plotted in the same sort order, but with the value of each successive topic added to the value of the preceding topic. The sorted means charts were used to identify the most significant topics (those at the top of the chart). The cumulative means charts were used to identify total exit level requirements within each of the computing programs. Data was also extracted from the sorted means charts and used to prepare tables of "top" exit level requirements for the three computing programs. Charts and tables are presented as Figures 12 and 13, and Tables 12, 13, 14, and 15 in the Results section.

4.2.6. Principle Component Plots

4.2.6.1. Most Important Topics

The graphical plot charts of means for responses (previously described) were visually examined to identify a breaking point for relatively "high" exit level responses for survey questions (The point for exit level 2.7 was selected.).[2] An imaginary "cut -off line" was drawn on charts at that point, and topics with "high" responses above that line were identified in tables showing the "most important" knowledge elements for each of the three programs. The charts are presented as Tables 9, 10, and 11 in the Results section.

[2] Exit level 2.7 was selected because visual inspection of all of the bar charts revealed this point divided a significant but also limited number of responses above that point on all charts, and a 2.7 exit level requires considerable knowledge (on a scale of 1 to 4).

4.2.6.2. Major Differences in Knowledge Elements

The delta charts (previously described) were visually examined to identify major areas of difference in responses to individual questions for the three programs. Tables of major differences (differences greater than ± .50) were prepared for each of the three comparison groups for which charts were developed. Data was analyzed and is presented in Tables 6, 7, and 8 in the Results section.

4.3. Analysis of Response Data to Verify Hypotheses

Survey results were analyzed using two methods: (1) inspection and comparison of survey response against anticipated results, and (2) statistical analysis.

4.3.1. Inspection and Comparison of Survey Response Data Against Anticipated Results

Differences, similarities, trends, "highs," and "lows" were identified in charts and tables of survey response data. These observations were compared with anticipated responses for each of the three computing programs surveyed, and results were analyzed to determine if the hypotheses are supported.

4.3.2 . Statistical Analysis of Survey Response Data

Data for statistical analysis of survey responses was developed in the following steps:

(1) Response data was entered on a Microsoft Excel® spreadsheet and sorted into the three response groups (CS, IS, and SE). Means and standard deviations were obtained for responses to each of the 68 topics within each of these groups.

(2) Means differences were obtained for responses to all topics within three comparison groups: IS - CS, IS - SE, and CS - SE (e.g., the mean for all responses to topic 1 for CS was subtracted from the mean for all responses to topic 1 for IS, producing a difference).

(3) Student's t-test technique was used to determine if the survey responses within each of the comparison groups (e.g., IS and CS for the IS - CS group) were "strongly different," were "not strongly different," or were "strongly the same" for the two groups.[3] The test produced "p-values."

(4) A formula was developed and used to convert the p-values obtained from the t-test to both positive and negative values, so that they could be sorted and charted into separate program groups for comparison. For a given topic in a means comparison group (e.g., IS - CS), if the topic of the first group's mean is larger, the mean difference will be positive. If the topic of the second group's mean is larger, the mean difference will be negative. This fact was used to develop the p-value conversion formula: [4]

"P" = (absolute value of D ÷ D) • (1.0 - p-value of D)

where:

"P" = The converted ± p-value, and

D = The difference in means for one topic of a comparison group (e.g., topic 1 for IS - CS).

(5) Comparison charts for means and "P"-values were developed. Means differences for each of the 68 topics were sorted (for descending means values) within each of the three comparison groups, and then plotted on bar charts. Using the same sort (based on topic), "P"-values were plotted on bar charts. Means and "P"-value charts are provided as Figures 14, 15, and 16 in the Results section of the thesis.

---

[3] Student's t-test examines two samples and produces a t-value. Normally, t-values can be looked up in a table to determine associated p-values. The p- value is a measure of the confidence that one sample is similar to or different from the other. Responses for the three comparison groups were examined using the Microsoft Excel® "TTEST" function for two-samples of unequal variance, with two distribution tails. This function skips the step of providing the t-value for a given two sample comparison, and provides an immediate p-value instead. High p-values for a topic indicate that the two samples have a high probability of being different for that topic. For this thesis, p-values equal to or greater than .95 were considered "high", and the compared samples were determined to be "strongly different". Likewise, comparisons with p-values equal to or less than .05 were considered to be "strongly the same". Comparisons having p-values between .05 and .95 were considered to represent "common" topics for which the two samples were not strongly different and also not strongly the same.

[4] Using this formula made it possible to chart p-values (converted to "P"-values) for one member of a comparison group above the chart's x-axis, and values of the other member of the group below the x-axis.

These charts show graphically how responses for topics for compared computing programs are "strongly different," "not strongly different," or "strongly the same."

(6) The charts developed in Step 5 do not contain topic description detail. Tables for topic descriptions and associated "P"-values for "strongly different" and "strongly the same" topics were developed and are provided in Tables 16, 17, and 18 in the Results section.

(7) The charts developed in Step 5 and the Tables developed in Step 6 do not contain topic description detail and associated "P"-values for "not strongly different" and "not strongly the same" topics. Data for this detail were developed and is provided in Tables 19, 20, 21, and 22 in the Results section.

5. RESULTS

5.1 Introduction

Research was directed at examining the extent to which survey participants accepted the body of knowledge for the three surveyed computing programs, and at establishing confidence levels for the extent to which they identified similar and different exit level expectations for elements of the body of knowledge.

The survey data set consisted of 140 responses. Each response was a record containing response ID, respondent program classification (IS, CS, SE), and the exit level responses to the 68 topic questions (See Appendix B). Topic means were computed for each program. These means were used to generate comparative statistics for presentation in charts and in tables. Charts were used to graphically represent significant differences and similarities between the programs. Tables were used to present word pictures of response topics.

5. 2. Response to the Survey

Persons responding to the survey identified themselves with one of the programs surveyed, computer science, information systems, or software engineering. Table 1 shows the number of persons responding for each of the programs. The mean number of questions answered was 67.4, out of 68 questions, with an overall standard deviation of 3.0. While the number of persons responding to the survey was small (140 out of approximately 1,000 persons surveyed), the standard deviation for the two smallest groups responding (computer science and software engineering) was minimal. Notably, the standard deviation for the smallest group responding (software engineering) was 0.0.

Table 1. Number of participants, mean, and standard deviation for survey participation

COMPUTING PROGRAM	NUMBER PARTICIPANTS	MEAN (NUMBER OF QUESTIONS ANSWERED)	STANDARD DEVIATION
CS	26	67.9	0.4
IS	108	67.2	3.4
SE	6	68.0	0.0
TOTAL	140	67.4	3.0

5.3 Analysis of Survey Response

Survey results were analyzed by examining response data for three response comparison groups: IS - CS, IS - SE, and CS - SE. These three groups cover all possible comparison possibilities.

In addition to charts and tables contained in this section, data is also provided in three appendices:

(1) Appendix A is a complete copy of the body of computing knowledge provided with the survey.

(2) Appendix B is a complete list of the 68 survey questions (topics) used in the survey.

(3) Appendix C is response statistical data, including counts, means, and standard deviations for survey responses. Data in Appendix C is provided for each of the computing programs for which data was received.

Many of the results are organized around the three groups of the common body of computing knowledge. These groups divide knowledge elements in the survey into broad categories, and will be referred to as Groups I, II, and III throughout the remainder of this thesis.

Group I. Corresponds to Questions 1 through 27 of the survey, and covers knowledge elements pertaining to "Information Technology." This group covers computer architectures, algorithms and data structures, programming languages, operating systems, telecommunications, database, and artificial intelligence. This part of the survey anticipated a strong CS response.

Group II. Corresponds to Questions 28 through 39 of the survey and covers "Organizational and Management Concepts." This group covers general organization theory, information systems management, decision theory, organizational behavior, managing the process of change, legal and ethical aspects of information systems, professionalism, and interpersonal skills. This part of the survey anticipated a strong IS response.

Group III. Corresponds to Questions 40 through 68 of the survey and covers "Theory and Development of Systems." This group covers systems and information concepts, approaches to systems development, systems development concepts and methodologies, systems development tools and techniques, applications planning, risk management, project management, information and business analysis, information systems design, systems implementation and testing strategies, systems operation and maintenance, and systems development for specific types of information systems. This part of the survey anticipated a strong SE response.

5.4. Order of Presentation and Analysis

Results are provided based on direct analysis of means for survey responses, as well as Student's t-test analysis of these responses. Results are presented and analyzed under eight headings:

(1) Pattern of responses (Section 5.5)

(2) Detailed means of responses (Section 5.6)

(3) High level comparisons (Section 5.7)

(4) Major differences ("Deltas") (Section 5.8)

(5) "Most important" topics (Section 5.9)

(6) Sorted and cumulative means comparisons (Section 5.10)

(7) "Strongly different" and "strongly the same" topics (Section 5.11)

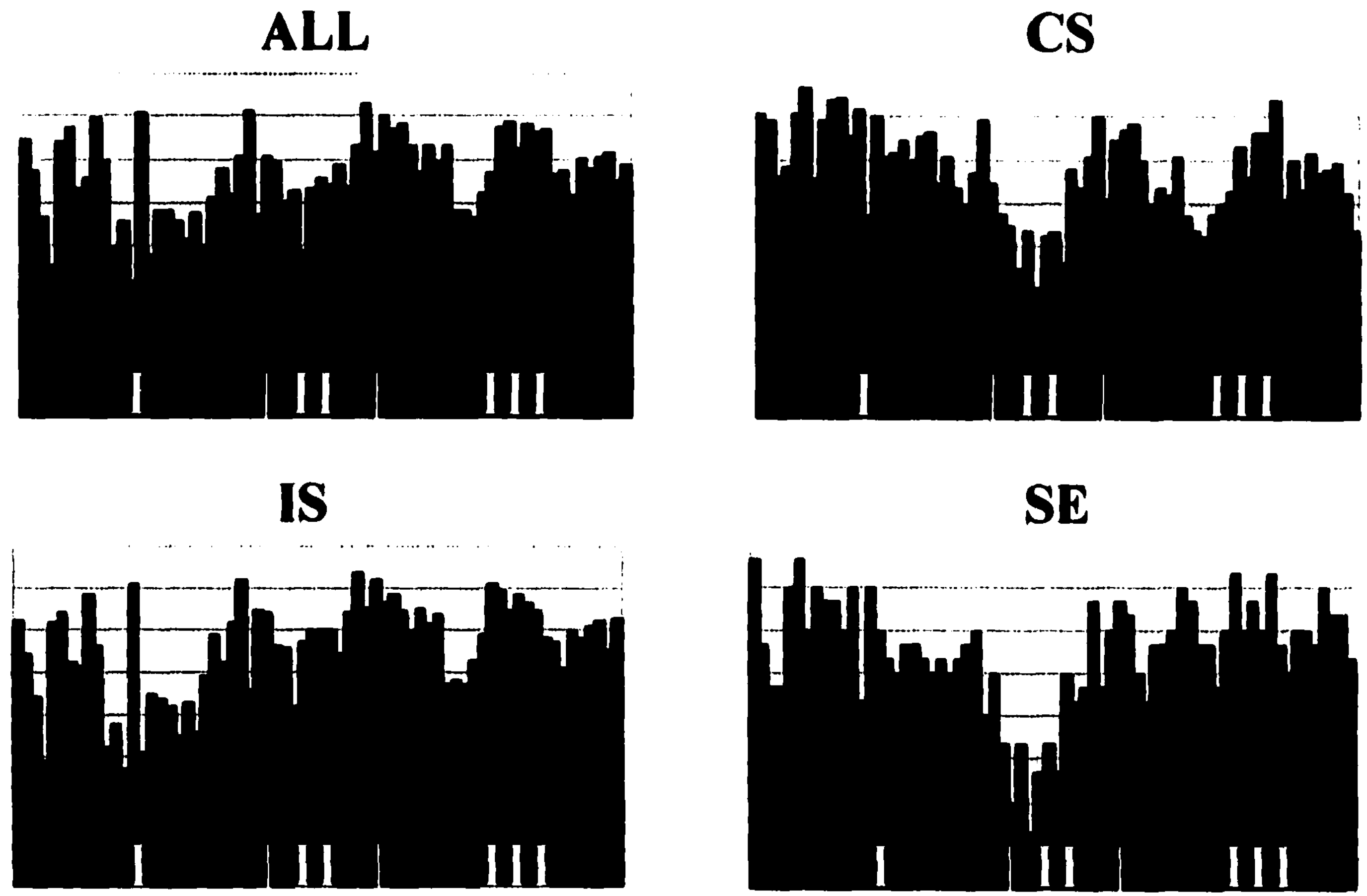
(8) "Not strongly different" and "not strongly the same" topics (Section 5.12)

5.5 Pattern of Responses

An overall visual impression of the similarities and differences in the total response to the survey can be obtained by examining the means response charts in Figure 8.

Each chart plots 68 vertical bars (one per topic) on the horizontal axis. Roman numerals represent the primary topic areas (1 - Information Technology, II - Organizational and Management Concepts, and III - Theory and Development of Systems). The vertical axis represents exit knowledge levels (0 - 4) for each topic.

The charts show that, while participants in each of the three computing programs accepted all elements of the body of knowledge, they placed significant differences on their exit level expectations for these elements. In general, without looking at any numbers, the following trends can be observed: (1) information systems expectations are higher in Group II and higher than CS in Group III, (2) computer science expectations are higher in Group I, lower than IS in Group II, and lower than SE in Group III. and (3) software engineering expectations are a little lower than CS in Group I, lower than CS in Group II, and higher than CS in Group III.



"I, II, and III" are the three primary levels of the common body of computing knowledge.

Figure 8. Means "patterns" for survey responses

5.6. Detailed Means of Responses

Figure 9 contains charts showing detailed means for the 68 survey questions, for each of the three computing programs, plus means for all responses. It provides the detail not provided in Figure 8, which contains only an overall visual representation. The charts show detailed means data. This same data is presented in different formats throughout other sections of the Results.

The charts show that the responses of participants were specific to the discipline with which they identified themselves, i.e.,

(1) Persons identifying themselves with CS placed emphasis on information technology and algorithm development knowledge elements more than on organization and management or development of systems;

(2) Persons identifying themselves with IS placed emphasis on knowledge elements pertaining to application of technology to organizations and on use of and development of systems more than on technology; and

(3) Persons identifying themselves with SE placed emphasis on use of technology and development of software more than technology or development of organizational systems.

Tables 2, 3, and 4 present means in the following exit knowledge ranges:

	Exit Knowledge Level Ranges
Table 2	3.00 - 3.25
Table 3	3.25 - 3.50
Table 4	3.50 - 4.00

By inspection of the tables, it is evident that each program respondent group places different importance on specified topic levels. Explanation of these differences will be examined in subsequent charts and tables. They are presented here to identify the fact there are differences in responses among the program respondents.

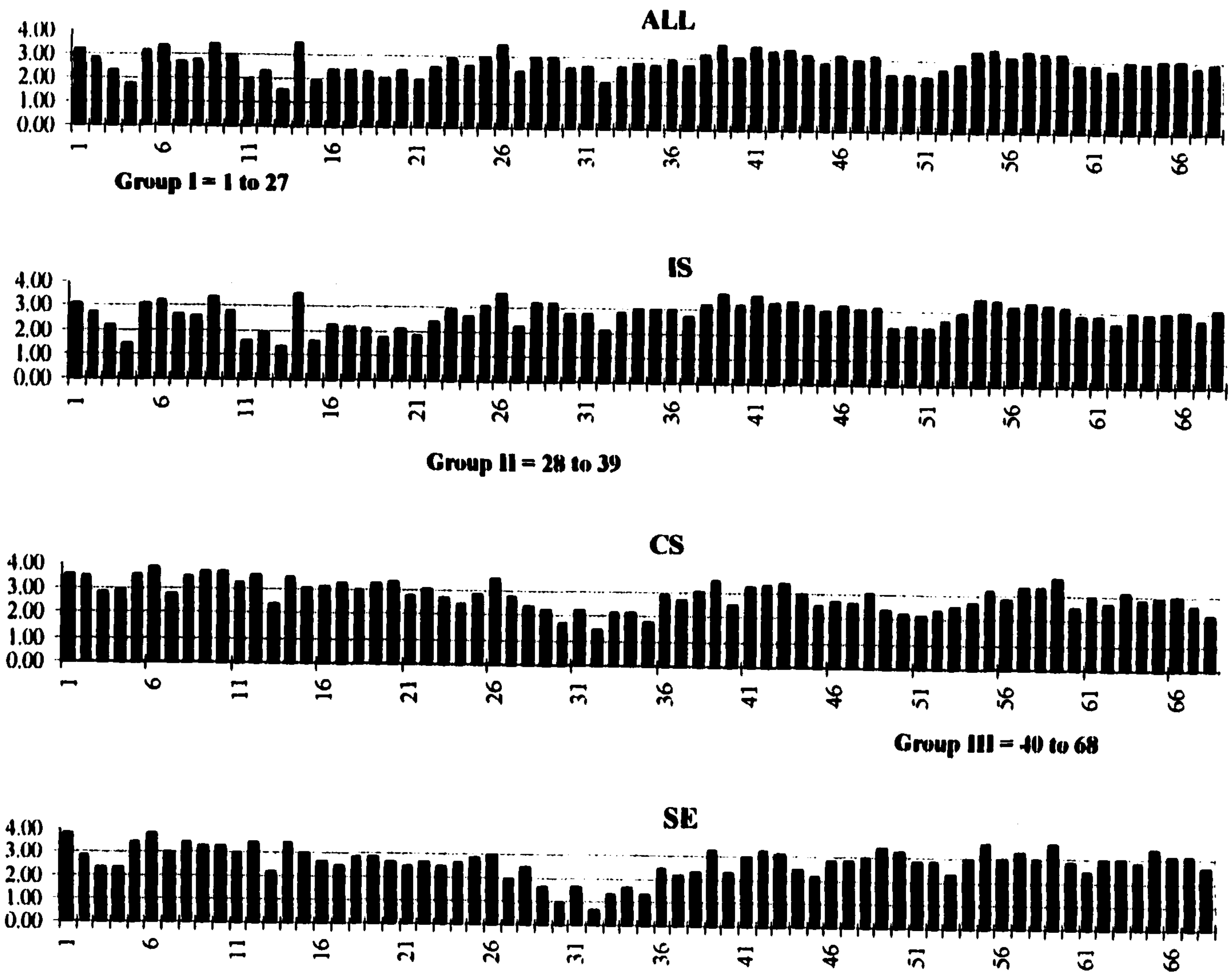


Figure 9. Detailed means charts for survey responsees

Table 2. Top "use level" exit levels for all programs, range 3.00 - 3.25

	RANGE	GROUP	TOPIC	IS	CS	SE	TOPIC DESCRIPTION
			1	3.12			Fundamental data presentation ...
			5	3.09			Formal problems and problem...
			6	3.22			Basic data structures
			7			3.00	Complex data structures
			11			3.00	Algorithm efficiency, complexity...
		I	15		3.04	3.00	Machine and assembly languages
			16		3.08		Design, implementation... languages...
			17		3.23		Architecture, goals... operating systems
			18		3		Interaction of operating system...
			22		3.04		Other operating system concerns
			25	3.09			Networks -- architectures...
			26			3.00	Database
			28	3.23			General organization theory
	3.00	II	29	3.22			Information systems management
U	to		35	3.01			Managing the process of change
S	3.25		38	3.22	3.04		Legal and ethical aspects
E			41		3.23	3.00	Approaches to systems development...
			43			3.17	Systems development concepts and...
			44	3.24	3		Systems development tools and...
			47	3.07			Project management (control)
			48	3.19	3.04	3.00	Project management (... documentation)...
			54			3.00	Information and business analysis
			55		3.15		Information systems design (logical...)
		III	56	3.23		3.00	Information systems design (human...)
			58			3.00	Information systems design (software...)
			59	3.23			Information systems design (software...)
			61		3		Information systems design (correctness...)
			62			3.00	Information systems design (verification...)
			63		3.08	3.00	Information systems design (software...)
			65	3.06			Information systems design(software...)
			66	3.11		3.17	Systems implementation and testing
			67			3.17	Systems operation and maintenance
			68	3.15			Systems development for specific...

Table 3. Top "intermediate level" exit levels for all programs, range 3.25 - 3.50

	RANGE	GROUP	TOPIC	IS	CS	SE	TOPIC DESCRIPTION
INTERMEDIATE	3.25 to 3.50	I	2		3.5		CPU architectures and computer system..
			8		3.5		Abstract data types
			9	3.4		3.33	File structures and access methods
			10			3.33	Sorting and searching data structures and...
			11		3.3		Algorithm efficiency, complexity and metrics
			19		3.3		Operating system process management
			20		3.3		Memory management
			26		3.5		Database
		II	39			3.33	Interpersonal skills -- oral and written...
		III	40	3.3			Systems and information concepts
			42	3.3	3.4	3.33	Approaches to systems development (systems...)...
			43	3.4	3.4		Systems development concepts and...
			46	3.3			Project management (organization...)...
			50			3.33	Configuration management (documentation)
			55	3.5			Information systems design (logical...).
			57	3.4	3.3	3.33	Information systems design (...requirements)...
			58	3.3	3.3		Information systems design (...specifications)...

Table 4. Top "application level" exit levels for all programs, range 3.50 - 4.00

	RANGE	GROUP	TOPIC	IS	CS	SE	TOPIC DESCRIPTION
APPLY	3.50 to 4.00	I	1		3.5	3.83	Fundamental data presentation and physical...
			5		3.5	3.50	Formal problems and problem solving
			6		3.9	3.83	Basic data structures
			8			3.50	Abstract data types
			9		3.7		File structures and access methods
			10		3.7		Sorting and searching data structures and...
			12		3.6	3.50	Recursive algorithms
			14	3.5	3.5	3.50	Programming languages
			26	3.6			Database
		II	39	3.7	3.5		Interpersonal skills
		III	41	3.6			Approaches to systems development (Models...)
			49			3.50	Configuration management (principles...)
			54	3.6			Information and business analysis
			55			3.67	Information systems design (logical...)
			59		3.7	3.67	Information systems design (software design)
			65			3.50	Information systems design(software testing)

5.7. High Level Comparisons

The IS'97 body of knowledge is a four-level hierarchy organized in three major groups:

Group I - Information Technology

Group II - Organizational and Management Concepts

Group III - Theory and Development of Systems

At the second level, each of the three groups is decomposed into an additional level. Table 5 is organized according to the major body of knowledge groups and shows the second level elements. Computationally, the topic means are aggregated into the second level means. That is, summary topic questions were aggregated for each second level groping.

Figure 10 contains means charts aggregated at level-2 of the body of computing knowledge. The 68 summary questions fall under 26 level-2 knowledge elements on the body of knowledge. Means for responses to the questions were averaged under the 26 level-2 elements, and plotted on charts for each of the computing programs. One chart was also prepared for all responses.

The level-2 charts have the advantage of leveling out spikes, up or down, for responses to individual questions. The charts show the following:

(1) IS is generally lower than CS and SE in Group I, higher than CS and SE in Group II, and higher than CS and SE in Group III.

(2) CS is higher than IS and SE in Group I. It is lower than IS but higher than SE in group II, and lower than IS, and mixed higher and lower than SE, in Group III.

(3) SE is generally higher than IS but lower than CS in Group I. It is lower than CS and IS in Group II, and mixed higher and lower than CS in Group III.

Table 5. Survey responses, aggregated at level-2 of the common body of computing knowledge

BK GROUP	LEV-2 NO.	CHART NO.	TOPIC (Numbers in parentheses are survey topic numbers)	CS	IS	SE	ALL
I.	1.1	1	Computer Architectures (1 - 4)	3.18	2.37	2.83	2.55
	1.2	2	Algorithms and Data Structures (5 - 13)	3.36	2.51	3.24	2.70
	1.3	3	Programming Languages (14 - 16)	3.21	2.44	3.06	2.61
	1.4	4	Operating Systems (17 - 22)	3.10	2.06	2.67	2.28
	1.5	5	Telecommunications (23 - 25)	2.67	2.89	2.67	2.84
	1.6	6	Database (26)	3.46	3.59	3.00	3.54
	1.7	7	Artificial Intelligence (27)	2.73	2.29	2.00	2.36
II.	2.1	8	General Organization Theory (28)	2.38	3.23	2.50	3.04
	2.2	9	Information Systems Management (29 - 32)	1.91	2.73	1.25	2.51
	2.3	10	Decision Theory (33)	2.12	2.85	1.33	2.65
	2.4	11	Organizational Behavior (34)	2.15	2.99	1.67	2.78
	2.7	12	Managing the Process of Change (35)	1.81	3.01	1.33	2.71
	2.8	13	Legal and Ethical Aspects of IS (36)	2.88	2.99	2.50	2.95
	2.9	14	Professionalism (37)	2.69	2.72	2.17	2.69
	2.10	15	Interpersonal Skills (38 - 39)	3.27	3.46	2.83	3.40
III.	3.1	16	Systems and Information Concepts (40)	2.54	3.26	2.33	3.09
	3.2	17	Approaches to Systems Development (41 - 42)	3.29	3.47	3.17	3.42
	3.3	18	Systems Development Concepts and Methodologies (43)	3.42	3.43	3.17	3.41
	3.4	19	Systems Development Tools and Techniques (44)	3.00	3.24	2.50	3.16
	3.5/3.6	20	Application Planning (and Risk Management) (45)	2.50	2.98	2.17	2.86
	3.7	21	Project Management (46 - 53)	2.48	2.78	2.94	2.73
	3.8	22	Information and Business Analysis (54)	2.64	3.55	3.00	3.36
	3.9	23	Information Systems Design (55 - 65)	3.02	3.09	3.12	3.08
	3.10	24	Systems Implementation and Testing Strategies (66)	2.96	3.11	3.17	3.09
	3.11	25	Systems Operation and Management (67)	2.62	2.75	3.17	2.75
	3.12	26	Systems Development for Specific Types of Information Systems (68)	2.19	3.15	2.67	2.95

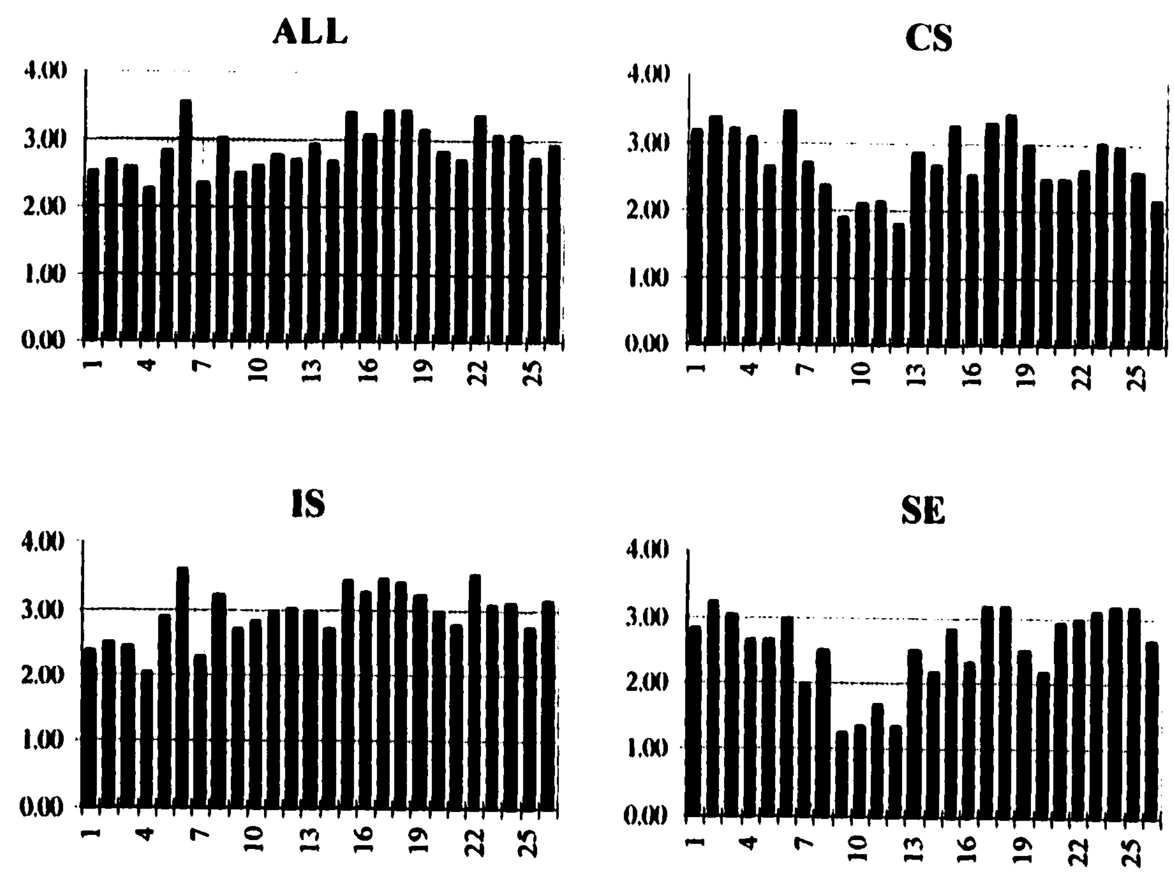


Figure 10. Survey responses aggregated at level-2 of the combined body of computing knowledge

5.8. Major Differences ("Deltas")

In order to examine differences between the three programs, the differences between the topic means were computed. The differences between means are referred to as "deltas."

Figure 11 shows differences (or "deltas") for responses to the 68 survey questions, for the three comparison groups. Actual means values are provided for all topics in Appendix C. The charts show:

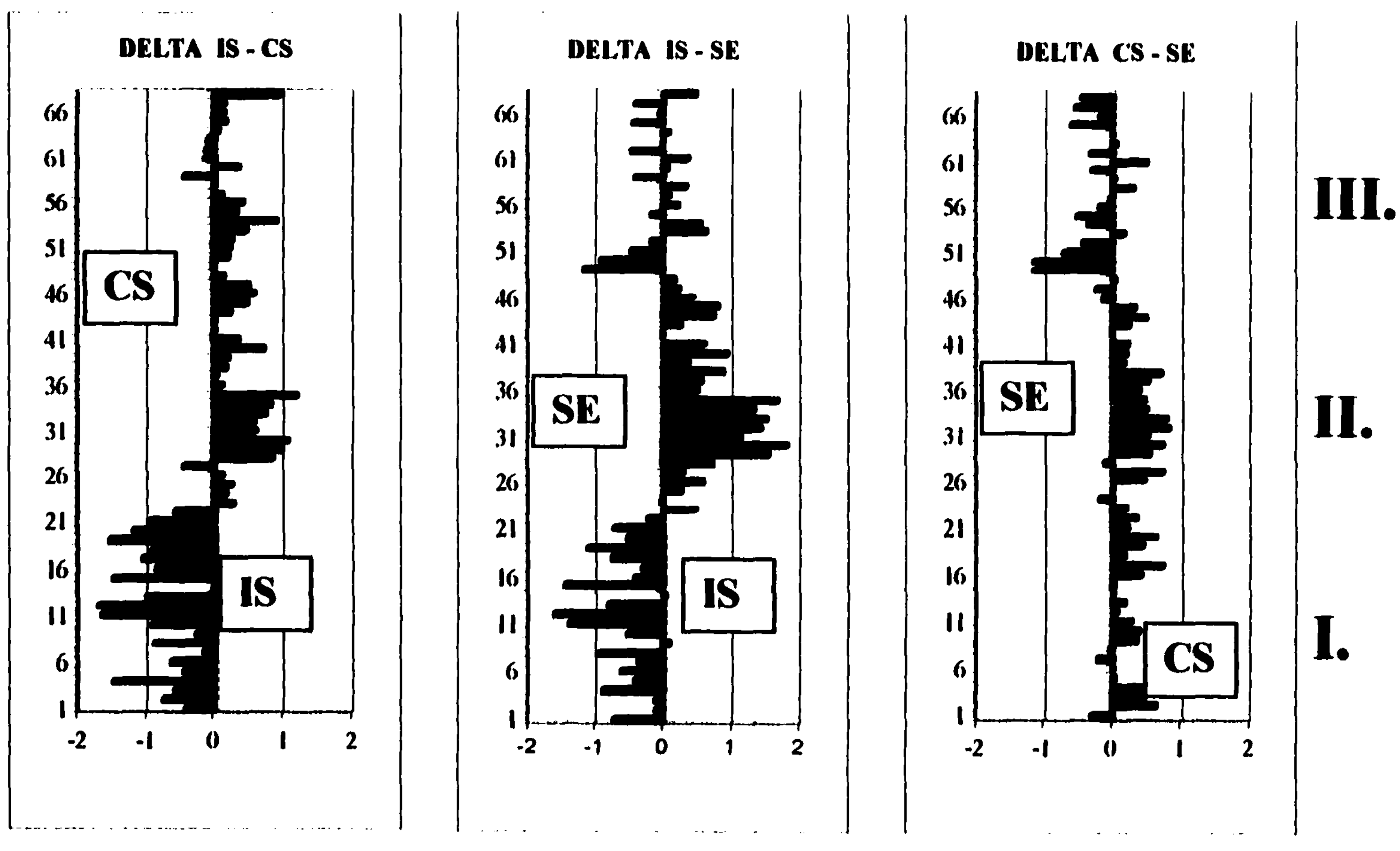
IS - CS Chart With several exceptions, IS has lower exit levels than CS in Group I. In Group I, IS is only slightly higher than CS in topic 14 (programming languages) and in topics 23, 24, and 25 (telecommunications and networks), and topic 26 (database). In Group II, IS is higher than CS for all topics, and in Group III, IS is generally higher than CS for most topics except 59, 61, 62, and 63 (all pertaining to information systems design).

IS - SE Chart. The differences between IS and SE are similar to those between IS and CS for Groups I and II. In Group I, SE is lower than IS in topics 9 (file structures), 14 (programming languages), 23 and 25(telecommunications and networks), 26 (database), and 27 (artificial intelligence). In Group III, IS is generally mixed higher and lower than SE, with topics 49 and 50 (pertaining to configuration management) significantly higher for SE. IS is significantly higher in Group III for topics 44 and 45 (pertaining to systems development).

CS - SE Chart. CS is consistently higher than SE in Groups I and II. SE is slightly higher in topics 1 (data presentation), 7 and 8 (data structures and data types), 24 (telecommunications), and 28 (organization theory). CS is lower than SE for most topics in Group III, especially for topics 49 and 50 (configuration management).

The data used to prepare the charts in Figure 11 was used to develop Tables 6, 7, and 8. The purpose of the tables is to show the topics associated with the most important differences between the three program groups. These tables show all topics with exit level differences at .50 and above and -.50 and below on the difference charts. On each table there are two groups of topics, those with negative difference values (without shading), and those with positive difference values (with shading). Topics without shading are more important to one program; topics with shading are important to another. For

example, on Table 6, the topics with shaded differences are more important to IS, and those without shading are more important to CS.



Group I = Topics 1 - 27; Group II = Topics 28 - 39; and Group III =Topics 40 - 68.

Figure 11 . Differences ("Deltas") for responses to the 68 survey topic questions

Table 6. Major differences between exit levels for IS and CS (Shading shows exit levels for topics more important to IS)

	NO.	IS	CS	DELTA IS-CS	QUESTION (TOPIC)
	2	2.72	3.46	-0.74	CPU architectures and computer system components
	3	2.21	2.81	-0.59	Multiprocessor architectures
	4	1.44	2.92	-1.48	Digital logic and systems
	6	3.22	3.85	-0.62	Basic data structures
	8	2.57	3.46	-0.90	Abstract data types
	10	2.81	3.72	-0.91	Sorting and searching data structures and algorithms
	11	1.61	3.27	-1.66	Algorithm efficiency, complexity and metrics
	12	1.89	3.58	-1.69	Recursive algorithms
I.	13	1.35	2.35	-1.00	Advanced consideration of algorithms
	15	1.55	3.04	-1.49	Machine and assembly languages
	16	2.23	3.08	-0.84	Design, implementation and comparison of programming languages
	17	2.18	3.23	-1.05	Architecture, goals, and structure of operating systems
	18	2.08	3.00	-0.92	Interaction of operating system and hardware architecture
	19	1.74	3.27	-1.53	Operating system process management
	20	2.13	3.31	-1.18	Memory management
	21	1.78	2.73	-0.95	Resource allocation and scheduling
	22	2.44	3.04	-0.59	Other operating system concerns
	28	3.23	2.38	0.84	General organization theory
	29	3.22	2.23	0.99	Information systems management
	30	2.81	1.73	1.08	Other information systems management -- staffing, human resource...
II.	31	2.79	2.19	0.60	Management of information systems sub-functions -- telecommunications...
	32	2.09	1.50	0.59	Computer operations management -- tape/DASD management, security...
	33	2.85	2.12	0.74	Decision theory -- measurement and modeling...
	34	2.99	2.15	0.84	Organizational behavior -- job design theory...
	35	3.01	1.81	1.20	Managing the process of change...
	40	3.26	2.54	0.73	Systems and information concepts -- general systems theory...
	46	3.27	2.69	0.58	Project management (organization and management)
III.	47	3.07	2.58	0.50	Project management (control)
	54	3.55	2.64	0.91	Information and business analysis
	68	3.15	2.19	0.96	Systems development for specific types of information systems

Table 7. Major differences between exit levels for IS and SE (Shading shows exit levels for topics more important to IS)

	NO.	IS	SE	DELTA IS-SE	QUESTION (TOPIC)
I.	1	3.12	3.83	-0.71	Fundamental data presentation and physical representation of ...
	4	1.44	2.33	-0.89	Digital logic and systems
	6	3.22	3.83	-0.61	Basic data structures
	8	2.57	3.50	-0.93	Abstract data types
	10	2.81	3.33	-0.52	Sorting and searching data structures and algorithms
	11	1.61	3.00	-1.39	Algorithm efficiency, complexity and metrics
	12	1.89	3.50	-1.61	Recursive algorithms
	13	1.35	2.17	-0.82	Advanced consideration of algorithms
	15	1.55	3.00	-1.45	Machine and assembly languages
	18	2.08	2.83	-0.75	Interaction of operating system and hardware architecture
	19	1.74	2.83	-1.10	Operating system process management
	20	2.13	2.67	-0.53	Memory management
	21	1.78	2.50	-0.72	Resource allocation an scheduling
	26	3.59	3.00	0.59	Database
II.	28	3.23	2.50	0.73	General organization theory
	29	3.22	1.67	1.55	Information systems management
	30	2.81	1.00	1.81	Other information systems management -- staffing, human resource...
	31	2.79	1.67	1.13	Management of information systems sub-functions -- telecommunications...
	32	2.09	0.67	1.43	Computer operations management -- tape/DASD management, security, etc.
	33	2.85	1.33	1.52	Decision theory -- measurement and modeling; group decision process...
	34	2.99	1.67	1.32	Organizational behavior -- job design theory, teamwork, ...
	35	3.01	1.33	1.68	Managing the process of change -- strategies for motivating change...
	37	2.72	2.17	0.56	Professionalism -- certification issues, current literature...
	38	3.22	2.33	0.89	Personal skills -- proactive behavior, goal setting, personal decision making...
III.	40	3.26	2.33	0.93	Systems and information concepts -- general systems theory...
	41	3.60	3.00	0.60	Approaches to systems development (models and techniques)
	44	3.24	2.50	0.74	Systems development tools and techniques -- CASE, Jackson techniques...
	45	2.98	2.17	0.81	Systems development application planning - hardware, database, security...
	49	2.35	3.50	-1.15	Configuration management (principles, concepts, roles)
	50	2.41	3.33	-0.93	Configuration management (documentation)
	53	2.96	2.33	0.63	Project tracking and close down
	54	3.55	3.00	0.55	Information and business analysis

Table 8. Major differences between exit levels for CS and SE (Shading shows exit levels for topics more important to CS)

	NO.	CS	SE	DELTA CS-SE	QUESTION (TOPIC)
I.	2	3.46	2.83	0.63	CPU architectures and computer system components
	4	2.92	2.33	0.59	Digital logic and systems
	17	3.23	2.50	0.73	Architecture, goals, and structure of operating systems
	20	3.31	2.67	0.64	Memory management
II.	27	2.73	2.00	0.73	Artificial intelligence
	29	2.23	1.67	0.56	Information systems management
	30	1.73	1.00	0.73	Other information systems management -- staffing, human resource...
	31	2.19	1.67	0.53	Management of information systems sub-functions -- telecommunications...
	32	1.50	0.67	0.83	Computer operations management -- tape/DASD management, security...
	33	2.12	1.33	0.78	Decision theory -- measurement and modeling; group decision process...
	37	2.69	2.17	0.53	Professionalism -- certification issues, current literature...
	38	3.04	2.33	0.71	Personal skills -- proactive behavior, goal setting, personal decision making...
III.	44	3.00	2.50	0.50	Systems development tools and techniques -- CASE, Jackson techniques...
	49	2.35	3.50	-1.15	Configuration management (principles, concepts, roles)
	50	2.19	3.33	-1.14	Configuration management (documentation)
	51	2.12	2.83	-0.71	Configuration management (organizational structures, plans, tools)
	55	3.15	3.67	-0.51	Information systems design (logical and physical design)
	61	3.00	2.50	0.50	Information systems design (software correctness and reliability)
	65	2.88	3.50	-0.62	Information systems design(software testing)
	67	2.62	3.17	-0.55	Systems operation and maintenance

5.9. "Most Important" Topics

The topics considered to be "most important" for each of the three surveyed computing programs are shown in Tables 9, 10, and 11. These tables were developed by visually examining the means charts in Figure 9 and drawing imaginary dividing lines across each chart. Inspection of these dividing lines revealed that lines drawn on all charts at exit level 2.70 appeared to split the bars plotted on the charts into a reasonable number of comparatively "high" and "low" bars.

Topics representing the "high" bar plots were then sorted in descending means order within each of the three primary body of knowledge groups, for each computing program. The results entered in the tables provide a picture of the most important topics in each program, divided into the three primary body of knowledge groups. The number of "most important" topics, by program, and by group is:

	Number of "most important" topics within each body of knowledge group		
	Group I	Group II	Group III
IS	10	11	24
CS	24	3	15
SE	16	1	23

Table 9. Most important IS Topics in Groups I, II, and III (Mean greater than or equal to 2.70)

	NO.	IS MEAN	QUESTION (TOPIC)
I.	26	3.59	Database
	14	3.54	Programming languages
	9	3.43	File structures and access methods
	6	3.22	Basic data structures
	1	3.12	Fundamental data presentation and physical representation of ...
	25	3.09	Networks -- architectures and protocols; LANs...
	5	3.09	Formal problems and problem solving
	23	2.96	Telecommunication -- international standards, models...
	10	2.81	Sorting and searching data structures and algorithms
	2	2.72	CPU architectures and computer system components
II.	39	3.69	Interpersonal skills -- oral and written communication, writing documentation..
	28	3.23	General organization theory
	38	3.22	Personal skills -- proactive behavior, goal setting, personal decision making...
	29	3.22	Information systems management
	35	3.01	Managing the process of change -- strategies for motivating change...
	36	2.99	Legal and ethical aspects -- software sales, licensing, contract fundamentals...
	34	2.99	Organizational behavior -- job design theory, teamwork, ...
	33	2.85	Decision theory -- measurement and modeling; group decision process...
	30	2.81	Other information systems management -- staffing, human resource...
	31	2.79	Management of information systems sub-functions -- telecommunications...
	37	2.72	Professionalism -- certification issues, current literature...
III.	41	3.60	Approaches to systems development (models and techniques)
	54	3.55	Information and business analysis
	55	3.49	Information systems design (logical and physical design)
	57	3.43	Information systems design (software requirements)
	43	3.43	Systems development concepts and methodologies -- data modeling...
	42	3.34	Approaches to systems development (systems engineering considerations)
	58	3.33	Information systems design (software specifications)
	46	3.27	Project management (organization and management)
	40	3.26	Systems and information concepts -- general systems theory...
	44	3.24	Systems development tools and techniques -- CASE, Jackson techniques...
	59	3.23	Information systems design (software design)
	56	3.23	Information systems design (human computer interaction)
	48	3.19	Project management (systems and user documentation)
	68	3.15	Systems development for specific types of information systems
	66	3.11	Systems implementation and testing strategies
	47	3.07	Project management (control)
	65	3.06	Information systems design (software testing)
	63	2.99	Information systems design (software implementation)
	45	2.98	Systems development application planning - hardware, database, security...
	53	2.96	Project tracking and close down
	64	2.92	Information systems design (software and hardware system integration)
	60	2.91	Information systems design (software quality assurance)
	61	2.86	Information systems design (software correctness and reliability)
	67	2.75	Systems operation and maintenance

Table 10. Most important CS topics in Groups I, II, and III (Mean greater than or equal to 2.70)

	NO.	CS MEAN	QUESTION (TOPIC)
	6	3.85	Basic data structures
	10	3.72	Sorting and searching data structures and algorithms
	9	3.69	File structures and access methods
	12	3.58	Recursive algorithms
	1	3.54	Fundamental data presentation and physical representation of ...
	5	3.54	Formal problems and problem solving
	14	3.5	Programming languages
	2	3.46	CPU architectures and computer system components
	8	3.46	Abstract data types
	26	3.46	Database
I	20	3.31	Memory management
	11	3.27	Algorithm efficiency, complexity and metrics
	19	3.27	Operating system process management
	17	3.23	Architecture, goals, and structure of operating systems
	16	3.08	Design, implementation and comparison of programming languages
	15	3.04	Machine and assembly languages
	22	3.04	Other operating system concerns
	18	3	Interaction of operating system and hardware architecture
	4	2.92	Digital logic and systems
	25	2.85	Networks -- architectures and protocols: LANs...
	3	2.81	Multiproessor architectures
	7	2.77	Complex data structures
	21	2.73	Resource allocation an scheduling
	27	2.73	Artificial intelligence
	39	3.5	Interpersonal skills -- oral and written communication, writing documentation..
II	38	3.04	Personal skills -- proactive behavior, goal setting, personal decision making...
	36	2.88	Legal and ethical aspects -- software sales, licensing, contract fundamentals...
	59	3.69	Information systems design (software design)
	43	3.42	Systems development concepts and methodologies -- data modeling...
	42	3.35	Approaches to systems development (systems engineering considerations)
	57	3.31	Information systems design (software requirements)
	58	3.31	Information systems design (software specifications)
	41	3.23	Approaches to systems development (models and techniques)
II	55	3.15	Information systems design (logical and physical design)
	63	3.08	Information systems design (software implementation)
	48	3.04	Project management (systems and user documentation)
	44	3	Systems development tools and techniques -- CASE, Jackson techniques...
	61	3	Information systems design (software correctness and reliability)
	66	2.96	Systems implementation and testing strategies
	65	2.88	Information systems design(software testing)
	64	2.85	Information systems design (software and hardware system integration)
	56	2.81	Information systems design (human computer interaction)

Table 11. Most important SE topics in Groups I, II, and III (Mean greater than or equal to 2.70)

	NO.	SE MEAN	QUESTION (TOPIC)
I.	1	3.83	Fundamental data presentation and physical representation of ...
	6	3.83	Basic data structures
	5	3.50	Formal problems and problem solving
	8	3.50	Abstract data types
	12	3.50	Recursive algorithms
	14	3.50	Programming languages
	9	3.33	File structures and access methods
	10	3.33	Sorting and searching data structures and algorithms
	7	3.00	Complex data structures
	11	3.00	Algorithm efficiency, complexity and metrics
	15	3.00	Machine and assembly languages
	26	3.00	Database
	2	2.83	CPU architectures and computer system components
	18	2.83	Interaction of operating system and hardware architecture
	19	2.83	Operating system process management
	25	2.83	Networks -- architectures and protocols: LANs...
II.	39	3.33	Interpersonal skills -- oral and written communication, writing documentation...
III.	55	3.67	Information systems design (logical and physical design)
	59	3.67	Information systems design (software design)
	49	3.50	Configuration management (principles, concepts, roles)
	65	3.50	Information systems design(software testing)
	42	3.33	Approaches to systems development (systems engineering considerations)
	50	3.33	Configuration management (documentation)
	57	3.33	Information systems design (software requirements)
	43	3.17	Systems development concepts and methodologies -- data modeling...
	66	3.17	Systems implementation and testing strategies
	67	3.17	Systems operation and maintenance
	41	3.00	Approaches to systems development (models and techniques)
	48	3.00	Project management (systems and user documentation)
	54	3.00	Information and business analysis
	56	3.00	Information systems design (human computer interaction)
	58	3.00	Information systems design (software specifications)
	62	3.00	Information systems design (verification/validation of software quality ...)
	63	3.00	Information systems design (software implementation)
	46	2.83	Project management (organization and management)
	47	2.83	Project management (control)
	51	2.83	Configuration management (organizational structures, plans, tools)
	52	2.83	Systems development quality assurance
	60	2.83	Information systems design (software quality assurance)
	64	2.83	Information systems design (software and hardware system integration)

5.10. Sorted and Cumulative Means Comparisons

Means data was sorted to explore distribution of exit level knowledge. Both means (Figure 12) and cumulative means (Figure 13) were plotted. Figure 12 presents data sorted on descending values of means. Figure 13 contains charts with the same sort order used for Figure 12, but with each successive means value added to the preceding, producing a cumulative means total.

Table 12 shows the data presented in Figure 12. Inspection of the charts and table shows which topics are most important to the respective programs.

The charts in Figure 13 show that the total of exit levels for all three computing programs is essentially the same, peaking at a total of just under 200 for all three programs.

Tables 13, 14, and 15 break out the data from Table 12 into the three program areas, and then reorganize the data within each knowledge level range. The data within each range shows topic material of significant importance to design of curricula within each program area.

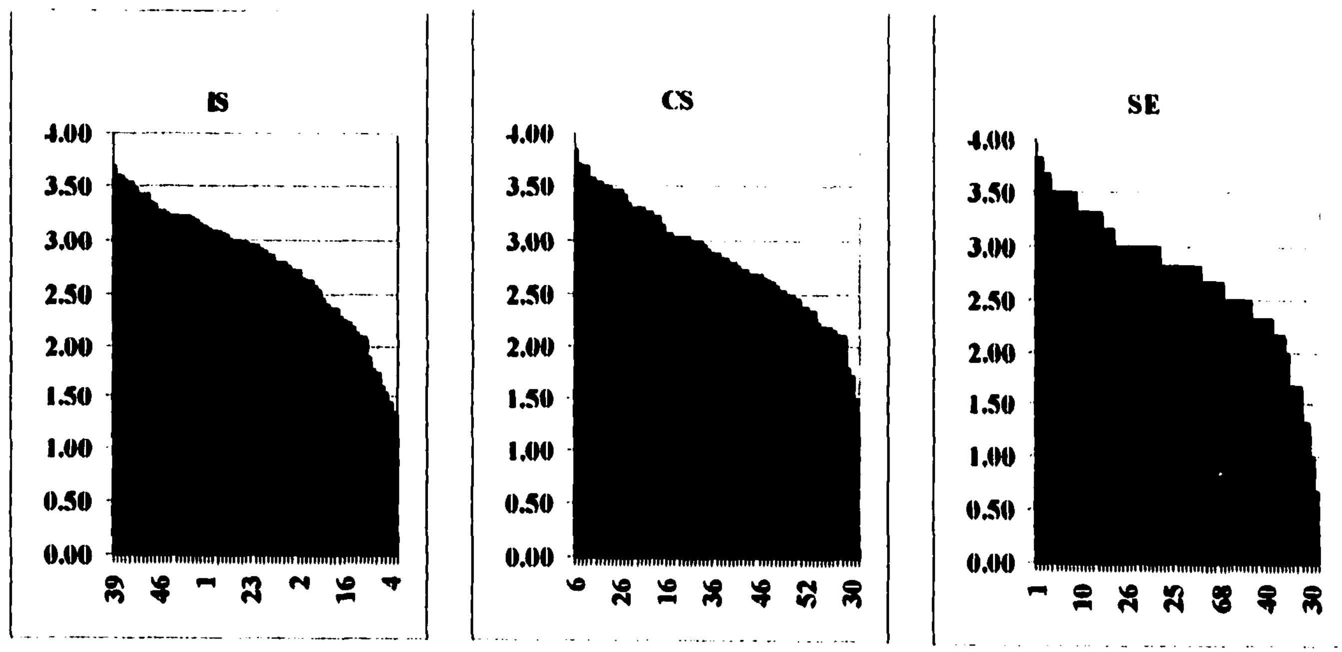


Figure 12. Topics, sorted on descending values of means

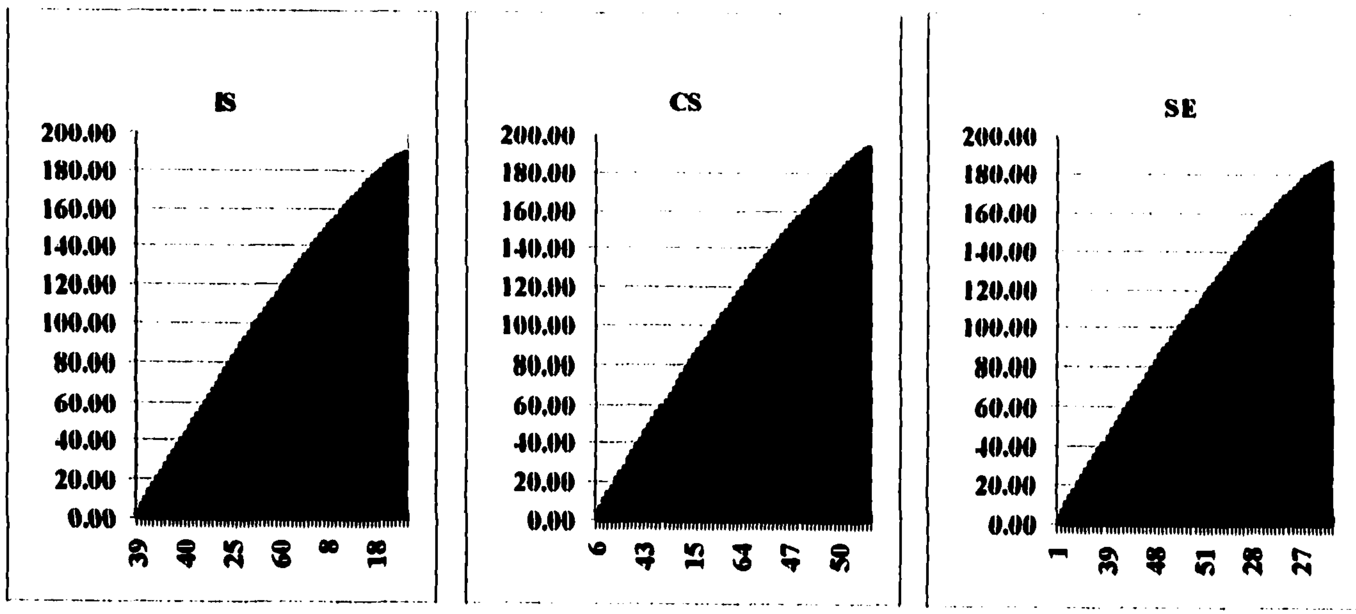


Figure 13. Topics, sorted on descending "cumulative" values of means

Table 12. Comparison of "top" exit level requirements

	EXIT LEVELS 3.00 OR ABOVE					
	TOPIC	CS	TOPIC	IS	TOPIC	SE
A (3.50)	6	3.85	39	3.69	1	3.83
	10	3.72	41	3.60	6	3.83
	9	3.69	26	3.59	55	3.67
	59	3.69	54	3.55	59	3.67
	12	3.58	14	3.54	5	3.50
	1	3.54	55	3.49	8	3.50
	5	3.54	57	3.43	12	3.50
	14	3.50	9	3.43	14	3.50
	39	3.50	43	3.43	49	3.50
B (3.25)	2	3.46	42	3.34	65	3.50
	8	3.46	58	3.33	9	3.33
	26	3.46	46	3.27	10	3.33
	43	3.42	40	3.26	39	3.33
	42	3.35	44	3.24	42	3.33
	20	3.31	59	3.23	50	3.33
	57	3.31	28	3.23	57	3.33
	58	3.31	56	3.23	43	3.17
	11	3.27	6	3.22	66	3.17
	19	3.27	38	3.22	67	3.17
C (3.00)	17	3.23	29	3.22	7	3.00
	41	3.23	48	3.19	11	3.00
	55	3.15	68	3.15	15	3.00
	16	3.08	1	3.12	26	3.00
	63	3.08	66	3.11	41	3.00
	15	3.04	25	3.09	48	3.00
	22	3.04	5	3.09	54	3.00
	38	3.04	47	3.07	56	3.00
	48	3.04	65	3.06	58	3.00
	18	3.00	35	3.01	62	3.00
	44	3.00			63	3.00
	61	3.00				
COUNT	31		29		30	

Table 13. Top IS exit levels, sorted by knowledge type, exit level range, group, topic, and exit level

	RANGE	GROUP	TOPIC	IS	TOPIC DESCRIPTION
APPLY	3.50 - 4.00	I	14	3.54	Programming languages
			26	3.59	Database
		II	39	3.69	Interpersonal skills -- oral and written communication, writing documentation..
		III	41	3.60	Approaches to systems development (models and techniques)
			54	3.55	Information and business analysis
INTERMED.	3.25 - 3.50	I	9	3.43	File structures and access methods
		III	40	3.26	Systems and information concepts -- general systems theory...
			42	3.34	Approaches to systems development (systems engineering considerations)
			43	3.43	Systems development concepts and methodologies -- data modeling...
			46	3.27	Project management (organization and management)
			55	3.49	Information systems design (logical and physical design)
			57	3.43	Information systems design (software requirements)
			58	3.33	Information systems design (software specifications)
USE	3.00 - 3.25	I	1	3.12	Fundamental data presentation and physical representation of ...
			5	3.09	Formal problems and problem solving
			6	3.22	Basic data structures
			25	3.09	Networks -- architectures and protocols; LANs...
		II	28	3.23	General organization theory
			29	3.22	Information systems management
			35	3.01	Managing the process of change -- strategies for motivating change...
			38	3.22	Personal skills -- proactive behavior, goal setting, personal decision making...
		III	44	3.24	Systems development tools and techniques -- CASE, Jackson techniques...
			47	3.07	Project management (control)
			48	3.19	Project management (systems and user documentation)
			56	3.23	Information systems design (human computer interaction)
			59	3.23	Information systems design (software design)
			65	3.06	Information systems design(software testing)
			66	3.11	Systems implementation and testing strategies
			68	3.15	Systems development for specific types of information systems

Table 14. Top CS exit levels, sorted by knowledge type, exit level range, group, topic, and exit level

	RANGE	GROUP	TOPIC	CS	TOPIC DESCRIPTION
APPLY	3.50 - 4.00	I	1	3.54	Fundamental data presentation and physical representation of ...
			5	3.54	Formal problems and problem solving
			6	3.85	Basic data structures
			9	3.69	File structures and access methods
			10	3.72	Sorting and searching data structures and algorithms
			12	3.58	Recursive algorithms
			14	3.50	Programming languages
		II	39	3.50	Interpersonal skills -- oral and written communication, writing documentation..
		III	59	3.69	Information systems design (software design)
INTERMED.	3.25 - 3.50	I	2	3.46	CPU architectures and computer system components
			8	3.46	Abstract data types
			11	3.27	Algorithm efficiency, complexity and metrics
			19	3.27	Operating system process management
			20	3.31	Memory management
			26	3.46	Database
		III	42	3.35	Approaches to systems development (systems engineering considerations)
			43	3.42	Systems development concepts and methodologies -- data modeling...
			57	3.31	Information systems design (software requirements)
			58	3.31	Information systems design (software specifications)
USE	3.00 - 3.25	I	15	3.04	Machine and assembly languages
			16	3.08	Design, implementation and comparison of programming languages
			17	3.23	Architecture, goals, and structure of operating systems
			18	3.00	Interaction of operating system and hardware architecture
			22	3.04	Other operating system concerns
		II	38	3.04	Personal skills -- proactive behavior, goal setting, personal decision making...
		III	41	3.23	Approaches to systems development (models and techniques)
			44	3.00	Systems development tools and techniques -- CASE, Jackson techniques...
			48	3.04	Project management (systems and user documentation)
			55	3.15	Information systems design (logical and physical design)
			61	3.00	Information systems design (software correctness and reliability)
			63	3.08	Information systems design (software implementation)

Table 15. Top SE exit levels, sorted by knowledge type, exit level, range, group, topic and exit level

	RANGE	GROUP	TOPIC	SE	TOPIC DESCRIPTION
APPLY	3.50 - 4.00	I	1	3.83	Fundamental data presentation and physical representation of ...
			5	3.50	Formal problems and problem solving
			6	3.83	Basic data structures
			8	3.50	Abstract data types
			12	3.50	Recursive algorithms
			14	3.50	Programming languages
		III	49	3.50	Configuration management (principles, concepts, roles)
			55	3.67	Information systems design (logical and physical design)
			59	3.67	Information systems design (software design)
			65	3.50	Information systems design(software testing)
INTERMED.	3.25 - 3.50	I	9	3.33	File structures and access methods
			10	3.33	Sorting and searching data structures and algorithms
		II	39	3.33	Interpersonal skills -- oral and written communication, writing..
		III	42	3.33	Approaches to systems development (systems engineering...)...
			50	3.33	Configuration management (documentation)
			57	3.33	Information systems design (software requirements)
USE	3.00 - 3.25	I	7	3.00	Complex data structures
			11	3.00	Algorithm efficiency, complexity and metrics
			15	3.00	Machine and assembly languages
			26	3.00	Database
		III	41	3.00	Approaches to systems development (models and techniques)
			43	3.17	Systems development concepts and methodologies -- data modeling...
			48	3.00	Project management (systems and user documentation)
			54	3.00	Information and business analysis
			56	3.00	Information systems design (human computer interaction)
			58	3.00	Information systems design (software specifications)
			62	3.00	Information systems design (verification/validation of software...)...
			63	3.00	Information systems design (software implementation)
			66	3.17	Systems implementation and testing strategies
			67	3.17	Systems operation and maintenance

5.11. <u>"Strongly Different" and "Strongly the Same" Topics</u>

Figures 14, 15, and 16 contain charts of sorted means differences or "deltas" (Chart 1), and "p" values (Chart 2), for topics within the three comparison groups (IS - CS, IS - SE, and CS - SE). Means difference values were plotted on the charts with their actual values. The "p" values, as explained below, are entered on the charts as "P" values, according to a conversion formula.

(1) <u>Chart 1, Figures 14, 15, and 16</u>. Means difference values that are positive all appear on the left side of the charts, above the x-axis. Means difference values that are negative appear on the right side of the charts, under the x-axis. Since the means difference values are sorted in descending order, this results in all of the values for one program being above the x-axis, and all of the values for the other program being below the axis. For example, in Chart 1 on Figure 14, all topics which have higher IS emphasis appear on the left side of the chart (above the x-axis), and all topics having a higher CS emphasis appear on the right side of the chart (below the x-axis). Topics plotted in the center of the chart have essentially the same value.

Topic numbers and actual means difference values are not shown on Chart 1 because the purpose of the chart is to provide a visual impression of how the means difference values for two compared programs are related. Actual topics and values are presented later, and the relevance of the charts becomes evident when Chart 2 is examined.

(2) <u>Chart 2, Figures 14, 15, and 16</u>. The p-values for means differences were plotted on Chart 2 as "P" values, using the conversion formula:

$$P = (\text{absolute value of } D \div D) \cdot (1.0 - \text{p-value of } D), \text{ where}$$

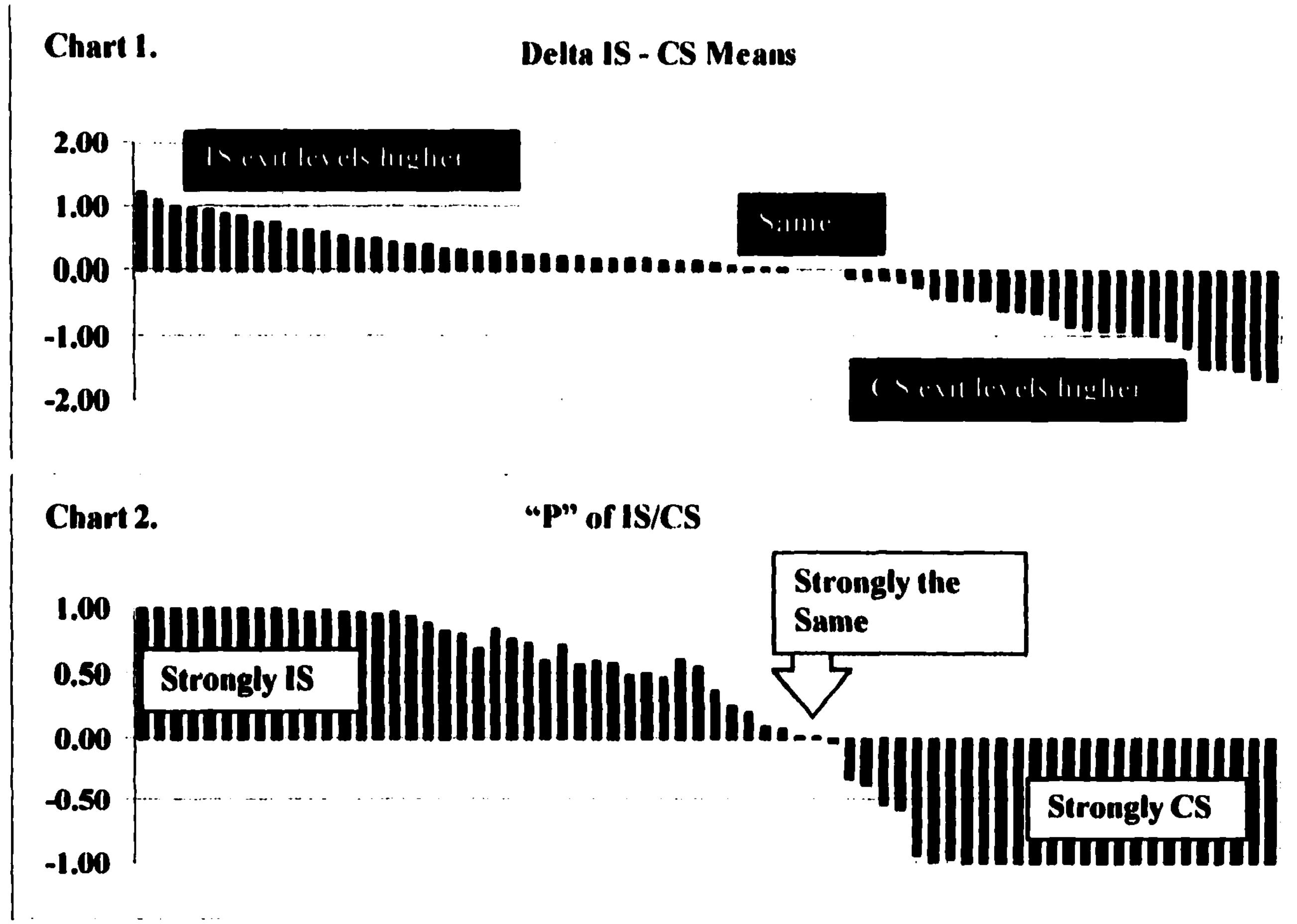
P = the converted ± p-value, and D = the difference in means for one topic of a comparison group (e.g., topic 1 for IS - CS).

Use of this formula enabled p-values to be plotted on both sides of the x-axis. Chart 2 on all figures contains converted p-values for means differences plotted in the same sort order as shown in Chart 1. What the charts permit is comparison of p-values for one program, compared with another, on two different sides of the x-axis. Topics with a high positive "P" value (.95 to 1.00) appear on the left side on top of the x-axis, and those with a high negative "P" value (-.95 to -1.00) appear on the right side of the chart on the bottom of the x-axis. Topics with similar values (0 to ± .05) appear in the middle of the chart, on both sides of the x-axis.

"P" values are a measure of the probability that one population sample is different from or the same as another. "High" values suggest that compared samples are different, and similar values suggest that they are the same. The significance of how these values can be used to compare programs can be illustrated by examination of one of the charts, Chart 2 on Figure 14. On this chart, topics labeled as "strongly IS" (plotted with "P" values of .95 to 1.0) all appear on the far left side of the chart, and topics labeled as "strongly CS" (plotted with "P" values of -.95 to -1.00) all appear on the right side. Topics labeled as "strongly the same" (plotted with "P" values between .05 and -.05) all appear in the center of the chart.

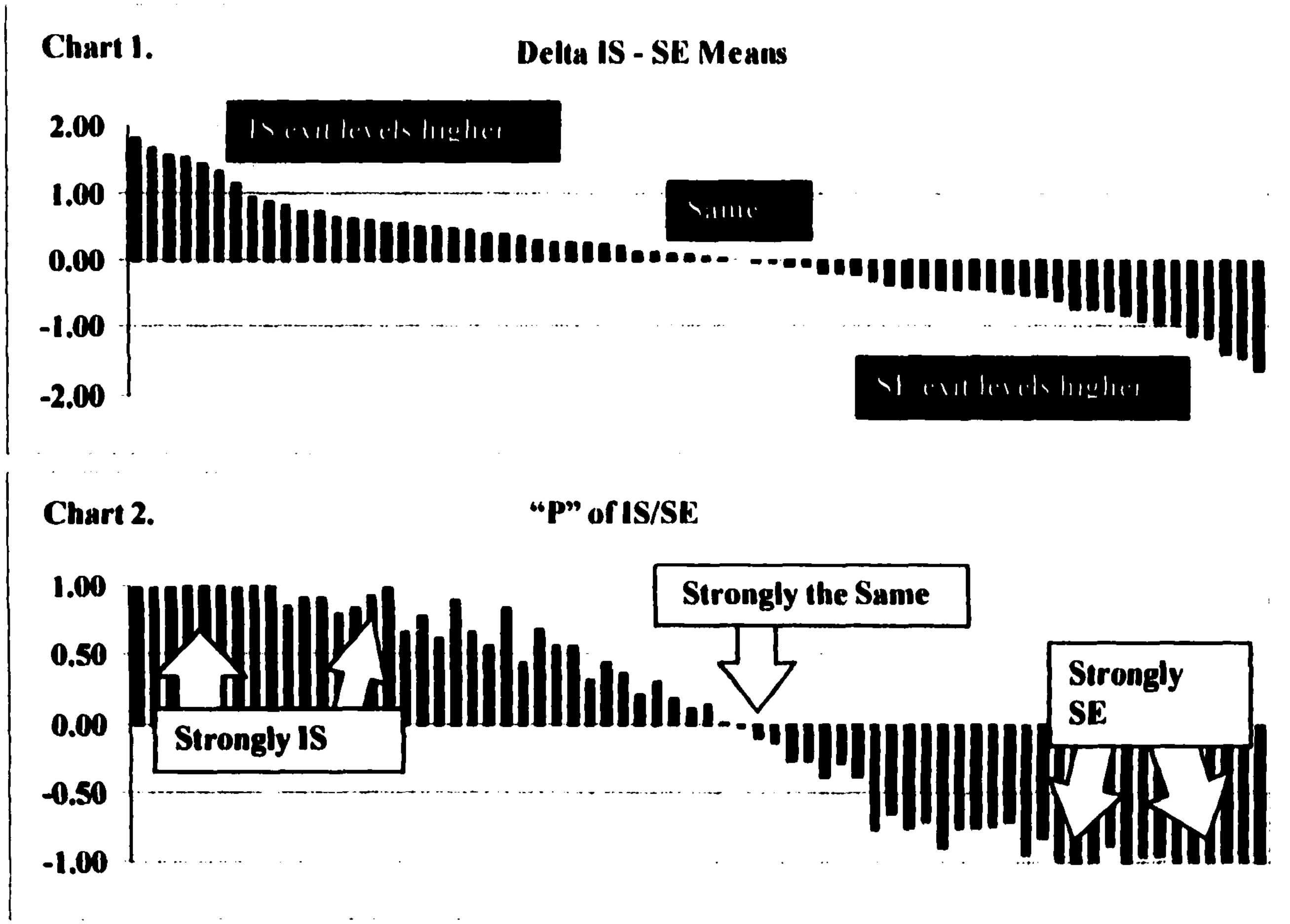
Topics labeled "strongly IS" are those which receive much more emphasis by IS than by CS. Likewise, those labeled "strongly CS" are those which receive much more emphasis by CS than by IS. Those labeled "strongly the same" are those which receive essentially the same emphasis by both IS and CS. Topics falling in between these ranges are "common" topics in the two programs. They are of course important, but they are not more strongly emphasized by either of the programs, when compared to each other.

Tables 16, 17, and 18 contain the detail for means differences and "P" values for the topics plotted on Figures 14, 15, and 16 as being strongly different or the strongly the same. Detail for "common" topics is provided in the next section.



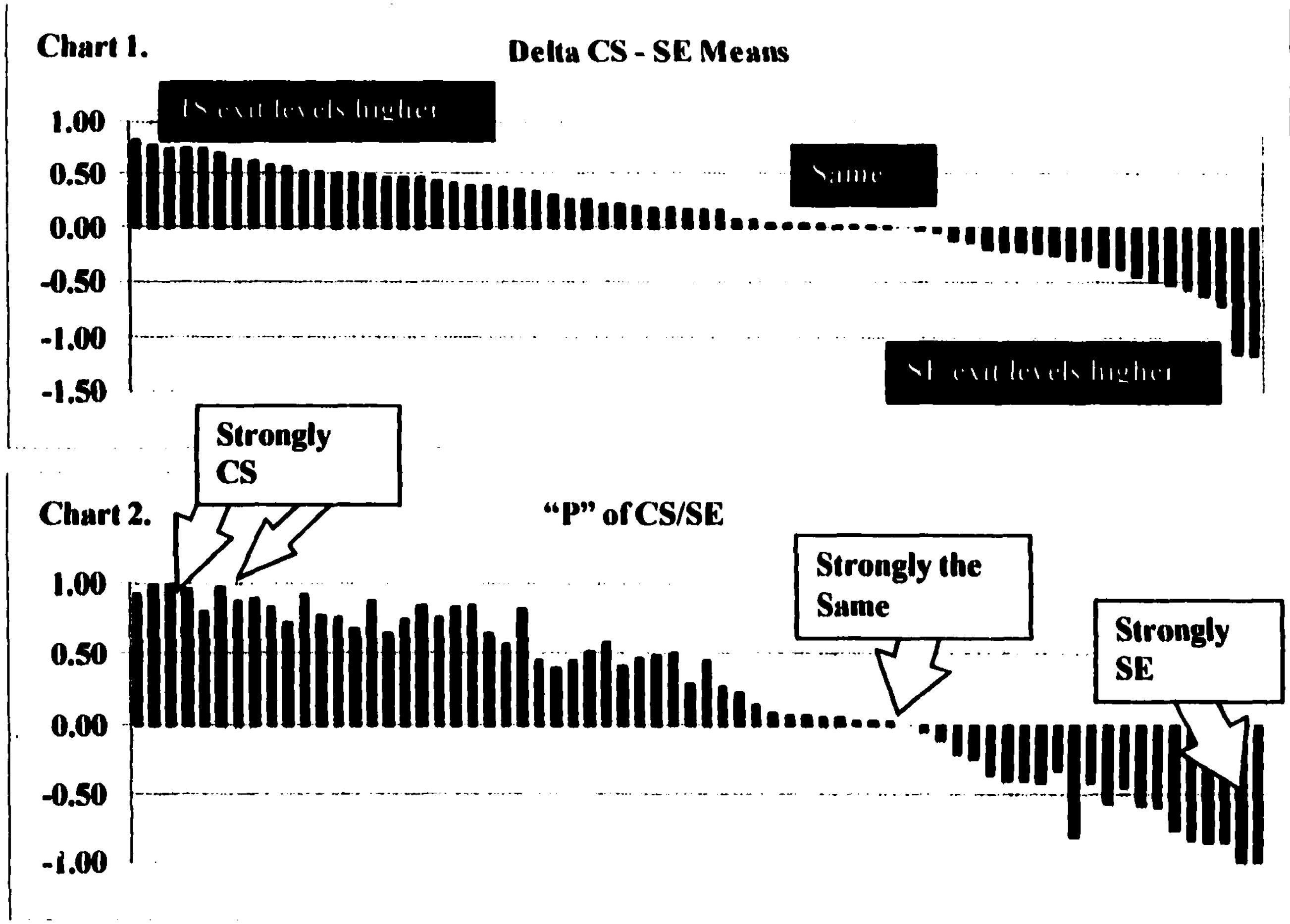
NOTE. Topic numbers and actual values were intentionally not provided in these charts. Detail is provided in other figures and tables.

Figure 14. Emphasis placed on survey topic exit levels by IS and CS survey participants, as shown by means differences and "P" values



NOTE. Topic numbers and actual values were intentionally not provided in these charts. Detail is provided in other figures and tables.

Figure 15. Emphasis placed on survey topic exit levels by IS and SE survey participants, as shown by means differences and "P" values



NOTE. Topic numbers and actual values were intentionally not provided in these charts. Detail is provided in other figures and tables.

Figure 16. Emphasis placed on survey topic exit levels by CS and SE survey participants, as shown by means differences and "P" values

Table 16. Probability of shared emphasis in topics between IS and CS

EMPHASIS	NO.	DELTA IS - CS MEAN	"P" of IS - CS	QUESTION (TOPIC)
STRONGLY IS	35	1.20	1.00	Managing the process of change -- strategies for motivating...
	30	1.08	1.00	Other information systems management -- staffing, human...
	29	0.99	1.00	Information systems management
	68	0.96	1.00	Systems development for specific types of information...
	54	0.91	1.00	Information and business analysis
	28	0.84	1.00	General organization theory
	34	0.84	1.00	Organizational behavior -- job design theory, teamwork...
	33	0.74	1.00	Decision theory -- measurement and modeling; group...
	40	0.73	0.99	Systems and information concepts -- general systems...
	31	0.60	0.98	Management of information systems sub-functions -- telecom...
	32	0.59	0.97	Computer operations management -- tape/DASD manage...
	46	0.58	0.99	Project management (organization and management)
	47	0.50	0.96	Project management (control)
	45	0.48	0.97	Systems development application planning - hardware...
	53	0.46	0.96	Project tracking and close down
	56	0.42	0.96	Information systems design (human computer interaction)
STRONGLY THE SAME	49	0.00	0.01	Configuration management (principles, concepts, roles)
	43	0.00	0.01	Systems development concepts and methodologies -- data...
	42	-0.01	-0.05	Approaches to systems development (systems engineering...)
STRONGLY CS	1	-0.42	-0.98	Fundamental data presentation and physical representation...
	27	-0.44	-0.96	Artificial intelligence
	5	-0.45	-0.99	Formal problems and problem solving
	59	-0.46	-1.00	Information systems design (software design)
	3	-0.59	-1.00	Multiproessor architectures
	22	-0.59	-1.00	Other operating system concerns
	6	-0.62	-1.00	Basic data structures
	2	-0.74	-1.00	CPU architectures and computer system components
	16	-0.84	-1.00	Design, implementation and comparison of programming...
	8	-0.90	-1.00	Abstract data types
	10	-0.91	-1.00	Sorting and searching data structures and algorithms
	18	-0.92	-1.00	Interaction of operating system and hardware architecture
	21	-0.95	-1.00	Resource allocation an scheduling
	13	-1.00	-1.00	Advanced consideration of algorithms
	17	-1.05	-1.00	Architecture, goals, and structure of operating systems
	20	-1.18	-1.00	Memory management
	4	-1.48	-1.00	Digital logic and systems
	15	-1.49	-1.00	Machine and assembly languages
	19	-1.53	-1.00	Operating system process management
	11	-1.66	-1.00	Algorithm efficiency, complexity and metrics
	12	-1.69	-1.00	Recursive algorithms

Table 17. Probability of shared emphasis in topics between IS and SE

INTEREST	NO.	DELTA IS - SE MEAN	"P" of IS - SE	QUESTION (TOPIC)
STRONGLY IS	30	1.81	0.99	Other information systems management -- staffing...
	35	1.68	0.99	Managing the process of change -- strategies for...
	29	1.55	0.99	Information systems management...
	33	1.52	1.00	Decision theory -- measurement and modeling; group...
	32	1.43	0.99	Computer operations management -- tape/DASD...
	34	1.32	1.00	Organizational behavior -- job design theory, teamwork ...
	31	1.13	0.98	Management of information systems sub-functions...
	40	0.93	1.00	Systems and information concepts -- general systems...
	38	0.89	0.99	Personal skills -- proactive behavior, goal setting...
	37	0.56	0.98	Professionalism -- certification issues, current literature...
STRONGLY THE SAME	42	0.00	0.01	Approaches to systems development (systems...)...
	63	-0.01	-0.02	Information systems design (software implementation)...
STRONGLY SE	6	-0.61	-0.99	Basic data structures...
	1	-0.71	-0.99	Fundamental data presentation and physical ...
	21	-0.72	-0.98	Resource allocation an scheduling...
	13	-0.82	-1.00	Advanced consideration of algorithms...
	4	-0.89	-0.96	Digital logic and systems...
	50	-0.93	-0.96	Configuration management (documentation)...
	8	-0.93	-0.99	Abstract data types...
	19	-1.10	-0.99	Operating system process management...
	49	-1.15	-0.98	Configuration management (principles, concepts, roles)...
	11	-1.39	-0.99	Algorithm efficiency, complexity and metrics...
	15	-1.45	-0.99	Machine and assembly languages...
	12	-1.61	-1.00	Recursive algorithms

Table 18. Probability of shared emphasis in topics between CS and SE

INTEREST	NO.	DELTA CS - SE MEAN	"P" of CS - SE	QUESTION (TOPIC)
STRONGLY CS	33	0.78	0.98	Decision theory -- measurement and modeling: group...
	17	0.73	0.98	Architecture. goals. and structure of operating sys...
	27	0.73	0.96	Artificial intelligence...
	38	0.71	0.97	Personal skills -- proactive behavior. goal setting...
STRONGLY THE SAME	6	0.01	0.05	Basic data structures...
	25	0.01	0.03	Networks -- architectures and protocols: LANs...
	42	0.01	0.03	Approaches to systems development (systems eng...)...
	64	0.01	0.03	Information systems design (software and hardware...)...
	14	0.00	0.00	Programming languages...
	57	-0.03	-0.05	Information systems design (software requirements)...
STRONGLY SE	50	-1.14	-0.98	Configuration management (documentation)...
	49	-1.15	-0.98	Configuration management (principles. concepts...)...

5.12. "Not Strongly Different" and "Not Strongly the Same" Topics

Tables 19, 20, and 21 are a follow-on to the charts and tables presented in the previous section. They identify topics which have "common" emphasis within each of the comparison groups. These tables contain all of the topics not contained on the previous charts for "strongly the same" or "strongly different."

Table 22 is a list of "common" topics which all of the surveyed computing programs appear to share in common, but which are not "strongly the same" or "strongly different" in any of them.

Table 19. Topics not strongly different between IS and CS

INTEREST	NO.	Delta IS - CS Means	"P" of IS/CS	QUESTION (TOPIC)
NOT STRONGLY DIFFERENT BETWEEN IS AND CS	7	-0.14	-0.58	Complex data structures
	9	-0.27	-0.94	File structures and access methods
	14	0.04	0.19	Programming languages
	23	0.27	0.80	Telecommunication -- international standards. models...
	24	0.16	0.58	Telecommunication -- bridges. routers. gateways...
	25	0.25	0.84	Networks -- architectures and protocols: LANs...
	26	0.13	0.61	Database
	36	0.11	0.37	Legal and ethical aspects -- software sales. licensing...
	37	0.03	0.09	Professionalism -- certification issues. current literature...
	38	0.18	0.56	Personal skills -- proactive behavior. goal setting. personal...
	39	0.19	0.72	Interpersonal skills -- oral and written communication..
	41	0.37	0.93	Approaches to systems development (models and...)...
	42	-0.01	-0.05	Approaches to systems development (systems...)...
	43	0.00	0.01	Systems development concepts and methodologies -- data...
	44	0.24	0.77	Systems development tools and techniques -- CASE...
	48	0.15	0.50	Project management (systems and user documentation)
	49	0.00	0.01	Configuration management (principles. concepts. roles)
	50	0.21	0.59	Configuration management (documentation)
	51	0.23	0.73	Configuration management (organizational structures...)...
	52	0.27	0.68	Systems development quality assurance
	55	0.33	0.83	Information systems design (logical and physical design)
	57	0.12	0.54	Information systems design (software requirements)
	58	0.02	0.07	Information systems design (software specifications)
	60	0.37	0.90	Information systems design (software quality assurance)
	61	-0.14	-0.54	Information systems design (software correctness and...)...
	62	-0.12	-0.37	Information systems design (verification/validation of...)...
	63	-0.09	-0.32	Information systems design (software implementation)
	64	0.07	0.24	Information systems design (software and hardware...)...
	65	0.17	0.59	Information systems design(software testing)
	66	0.15	0.49	Systems implementation and testing strategies
	67	0.14	0.46	Systems operation and maintenance

Table 20. Topics not strongly different between IS and SE

INTEREST	NO.	Delta IS - SE Means	"P" of IS/SE	QUESTION (TOPIC)
NOT STRONGLY DIFFERENT BETWEEN IS AND SE	2	-0.11	-0.26	CPU architectures and computer system components...
	3	-0.12	-0.26	Multiprocessor architectures...
	5	-0.41	-0.71	Formal problems and problem solving...
	7	-0.37	-0.64	Complex data structures...
	9	0.09	0.30	File structures and access methods...
	10	-0.52	-0.94	Sorting and searching data structures and algorithms...
	14	0.04	0.13	Programming languages...
	16	-0.43	-0.89	Design, implementation and comparison of programming...
	17	-0.32	-0.77	Architecture, goals, and structure of operating...
	18	-0.75	-0.88	Interaction of operating system and hardware...
	20	-0.53	-0.82	Memory management...
	22	-0.22	-0.37	Other operating system concerns...
	23	0.46	0.90	Telecommunication -- international standards, models...
	24	-0.04	-0.10	Telecommunication -- bridges, routers, gateways...
	25	0.26	0.55	Networks -- architectures and protocols: LANs...
	26	0.59	0.93	Database...
	27	0.29	0.67	Artificial intelligence...
	28	0.73	0.91	General organization theory...
	36	0.49	0.78	Legal and ethical aspects -- software sales, licensing...
	39	0.36	0.84	Interpersonal skills -- oral and written communication..
	41	0.60	0.84	Approaches to systems development (models and...)...
	42	0.00	0.01	Approaches to systems development (systems...)...
	43	0.26	0.55	Systems development concepts and methodologies...

Table 20, continued

INTEREST	NO.	Delta IS - SE Means	"P" of IS/SE	QUESTION (TOPIC)
NOT STRONGLY DIFFERENT BETWEEN IS AND SE	44	0.74	0.92	Systems development tools and techniques -- CASE...
	45	0.81	0.85	Systems development application planning - hardware...
	46	0.44	0.67	Project management (organization and management)...
	47	0.24	0.32	Project management (control) ...
	48	0.19	0.36	Project management (systems and user documentation)...
	51	-0.48	-0.71	Configuration management (organizational structures...)...
	52	-0.18	-0.28	Systems development quality assurance... -
	53	0.63	0.80	Project tracking and close down...
	54	0.55	0.66	Information and business analysis...
	55	-0.18	-0.38	Information systems design (logical and physical...)...
	56	0.23	0.43	Information systems design (human computer...)
	57	0.10	0.21	Information systems design (software requirements)...
	58	0.33	0.44	Information systems design (software specifications)...
	59	-0.43	-0.74	Information systems design (software design)...
	60	0.07	0.11	Information systems design (software quality assurance)...
	61	0.36	0.55	Information systems design (software correctness...)...
	62	-0.46	-0.73	Information systems design (verification/validation...)...
	63	-0.01	-0.02	Information systems design (software implementation)
	64	0.08	0.19	Information systems design (software and hardware...)
	65	-0.44	-0.74	Information systems design(software testing)...
	66	-0.05	-0.13	Systems implementation and testing strategies...
	67	-0.41	-0.75	Systems operation and maintenance...
	68	0.48	0.62	Systems development for specific types of information.

Table 21. Topics not strongly different between CS and SE

INTEREST	NO.	Delta CS - SE Means	"P" of CS/SE	QUESTION (TOPIC)
NOT STRONGLY DIFFERENT BETWEEN CS AND SE	1	-0.29	-0.80	Fundamental data presentation and physical ...
	2	0.63	0.89	CPU architectures and computer system
	3	0.47	0.76	Multiproessor architectures
	4	0.59	0.84	Digital logic and systems
	5	0.04	0.08	Formal problems and problem solving
	6	0.01	0.05	Basic data structures
	7	-0.23	-0.42	Complex data structures
	8	-0.04	-0.11	Abstract data types
	9	0.36	0.83	File structures and access methods
	10	0.39	0.85	Sorting and searching data structures and
	11	0.27	0.47	Algorithm efficiency. complexity and metrics
	12	0.08	0.22	Recursive algorithms
	13	0.18	0.50	Advanced consideration of algorithms
	14	0.00	0.00	Programming languages
	15	0.04	0.07	Machine and assembly languages
	16	0.41	0.84	Design. implementation and comparison of programming...
	18	0.17	0.29	Interaction of operating system and hardware
	19	0.44	0.77	Operating system process management
	20	0.64	0.88	Memory management
	21	0.23	0.59	Resource allocation an scheduling
	22	0.37	0.57	Other operating system concerns
	23	0.19	0.48	Telecommunication -- international standards. models...
	24	-0.21	-0.40	Telecommunication -- bridges. routers. gateways...
	25	0.01	0.03	Networks -- architectures and protocols: LANs...
	26	0.46	0.85	Database
	28	-0.12	-0.22	General organization theory
	29	0.56	0.72	Information systems management
	30	0.73	0.81	Other information systems management -- staffing...
	31	0.53	0.78	Management of information systems sub-functions...
	32	0.83	0.93	Computer operations management -- tape/DASD...
	34	0.49	0.88	Organizational behavior -- job design theory. teamwork. ...

Table 21, continued

INTEREST	NO.	Delta CS - SE Means	"P" of CS/SE	QUESTION (TOPIC)
	35	0.47	0.65	Managing the process of change -- strategies for...
	36	0.38	0.65	Legal and ethical aspects -- software sales, licensing...
	37	0.53	0.93	Professionalism -- certification issues, current literature...
	39	0.17	0.46	Interpersonal skills -- oral and written communication..
	40	0.21	0.47	Systems and information concepts -- general systems...
	41	0.23	0.41	Approaches to systems development (models...)
	42	0.01	0.03	Approaches to systems development (systems...)...
	43	0.26	0.52	Systems development concepts and methodologies...
	44	0.50	0.77	Systems development tools and techniques -- CASE...
	45	0.33	0.46	Systems development application planning - hardware...
	46	-0.14	-0.24	Project management (organization and management)
	47	-0.26	-0.33	Project management (control)
NOT	48	0.04	0.07	Project management (systems and user documentation)
STRONGLY	51	-0.71	-0.85	Configuration management (organizational structures...)...
DIFFERENT	52	-0.45	-0.57	Systems development quality assurance
BETWEEN	53	0.17	0.27	Project tracking and close down
CS AND SE	54	-0.36	-0.45	Information and business analysis
	55	-0.51	-0.77	Information systems design (logical and physical...)...
	56	-0.19	-0.35	Information systems design (human computer...)...
	57	-0.03	-0.05	Information systems design (software requirements)
	58	0.31	0.40	Information systems design (software specifications)
	59	0.03	0.06	Information systems design (software design)
	60	-0.29	-0.41	Information systems design (software quality assurance)...
	61	0.50	0.69	Information systems design (software correctness...)...
	62	-0.35	-0.56	Information systems design (verification/validation...)...
	63	0.08	0.14	Information systems design (software implementation)...
	64	0.01	0.03	Information systems design (software and hardware...)...
	65	-0.62	-0.85	Information systems design(software testing)
	66	-0.21	-0.41	Systems implementation and testing strategies
	67	-0.55	-0.84	Systems operation and maintenance
	68	-0.47	-0.59	Systems development for specific types of information...

Table 22. Topics for which interests are not strongly different between all programs

NOT STRONGLY DIFFERENT BETWEEN ALL PROGRAMS							
	Delta (Means)			"P" of:			
NO.	IS - CS	IS - SE	CS - SE	IS/CS	IS/SE	CS/SE	QUESTION (TOPIC)
7	-0.14	-0.37	-0.23	-0.58	-0.64	-0.42	Complex data structures...
9	-0.27	0.09	0.36	-0.94	0.30	0.83	File structures and access methods...
14	0.04	0.04	0.00	0.19	0.13	0.00	Programming languages...
23	0.27	0.46	0.19	0.80	0.90	0.48	Telecommunication -- international...
24	0.16	-0.04	-0.21	0.58	-0.10	-0.40	Telecommunication -- bridges, routers...
25	0.25	0.26	0.01	0.84	0.55	0.03	Networks -- architectures and protocols...
26	0.13	0.59	0.46	0.61	0.93	0.85	Database
36	0.11	0.49	0.38	0.37	0.78	0.65	Legal and ethical aspects -- software...
39	0.19	0.36	0.17	0.72	0.84	0.46	Interpersonal skills -- oral and written...
41	0.37	0.60	0.23	0.93	0.84	0.41	Approaches to systems dev... (models...)
42	-0.01	0.00	0.01	-0.05	0.01	0.03	Approaches to systems dev...(systems...)
43	0.00	0.26	0.26	0.01	0.55	0.52	Systems development concepts and...
44	0.24	0.74	0.50	0.77	0.92	0.77	Systems development tools...
48	0.15	0.19	0.04	0.50	0.36	0.07	Project management (systems and user...)
51	0.23	-0.48	-0.71	0.73	-0.71	-0.85	Configuration management (organiza...)
52	0.27	-0.18	-0.45	0.68	-0.28	-0.57	Systems development quality...
55	0.33	-0.18	-0.51	0.83	-0.38	-0.77	Info sys design (logical and...)
57	0.12	0.10	-0.03	0.54	0.21	-0.05	Info sys design (software requirements)
58	0.02	0.33	0.31	0.07	0.44	0.40	Info sys design (software spec...)
60	0.37	0.07	-0.29	0.90	0.11	-0.41	Info sys design (software quality...)
61	-0.14	0.36	0.50	-0.54	0.55	0.69	Info sys design (software correctness...)
62	-0.12	-0.46	-0.35	-0.37	-0.73	-0.56	Info sys design (verification/validation...)
63	-0.09	-0.01	0.08	-0.32	-0.02	0.14	Info sys design (software imple...)
64	0.07	0.08	0.01	0.24	0.19	0.03	Info sys design (software and hardware...)
65	0.17	-0.44	-0.62	0.59	-0.74	-0.85	Info sys design (software testing)
66	0.15	-0.05	-0.21	0.49	-0.13	-0.41	Systems implementation and testing...
67	0.14	-0.41	-0.55	0.46	-0.75	-0.84	Systems operation and maintenance...

6. DISCUSSION OF RESULTS

The thesis investigates the hypotheses that survey participants accept a common body of computing knowledge, and that they place both similar and different emphasis on key topics of the body of knowledge based on their identification with one of the three surveyed computing programs. If they do accept the body of knowledge and identify significant anticipated differences and similarities within the programs, this would have significant impact on future curriculum development efforts for the three programs. Exploring the hypotheses requires analysis of survey data to address the following issues:

1. Were all aspects of the body of computing knowledge acceptable to all survey participants regardless of their program classification?

The body of knowledge appears to be acceptable to all survey participants since they almost universally assigned exit levels to the knowledge elements in the body of knowledge, they did not reject any of the elements, and they did not offer any new elements. This observation is supported by answers to several questions:

(1) Was the survey instrument valid?

The survey was a pilot effort, this being the first attempt to survey computing professionals in computer science, information systems, and software engineering concerning their acceptance of a common body of knowledge for the three programs. The validity of the survey instrument can be established thorough its usefulness in gathering initial information concerning acceptance of the body of knowledge and identification of knowledge exit levels. The fact is that most of the survey participants answered most of the questions (See Table 1.), and none of the participants submitted additional elements for consideration in the body of knowledge

(2) Were the survey knowledge elements and exit levels recognized?

Participants indicated they understood the content of the summary questions and exit level designations by their ability to answer the questions, and by the consistency of their answers within their program groups. Participants used the knowledge levels, including the response "no assumed knowledge is relevant," in completing the survey. Also, there were no complaints regarding comprehension of the knowledge levels.

(3) Was the survey sample valid?

The survey instrument was identified as the, "IS'95 Survey of Curriculum Issues - Computer Science, Information Systems, and Software Engineering." Of approximately 1,000 persons surveyed, 140 responded. This included 108 for information systems, 26 for computer science, and only 6 for software engineering. This imbalance in survey participation between the three computing programs raises some concerns:

One concern is that the survey originated as an information systems survey effort, and the documentation that accompanied the survey instructions (including the common body of computing knowledge) was labeled as pertaining largely to "information systems." It is certainly possible that potential survey participants became confused when reviewing these materials, and declined to participate based on their assumption that the survey pertained primarily to information systems professionals.

Another concern is that the primary source of participants was lists of attendees at meetings of professional organizations, and most of these meetings and professional organizations were associated with either information systems or computer science. While the software engineering response to the survey is helpful as a starting point for comparison with information systems and computer science, the response is probably inadequate for use in developing substantial conclusions. An unsuccessful attempt was made to get responses from software engineers, and approximately 100 names of potential reviewers were obtained from the Software Engineering Institute at Carnegie Mellon University. However, only 6 persons identifying themselves as software engineers responded to the survey.

2. <u>Did survey results reveal significant anticipated similarities and differences in the three programs surveyed (sufficient to identify them as separate computing programs, with each having emphasis on significantly different and specific topics)</u>?

A major proposition of the hypothesis is that there are significant similarities and differences between the three computing programs IS, CS, and SE. Similarities and differences between the programs were determined in two ways:

(1) Testing (the Student T-test) of differences between the means of each discipline revealed that there are may elements (See Figures 14, 15, and 16.) which are strongly different ($p < .05$), as well as many which are strongly the same, between the programs. By inspection of the panels of phrases which describe those topics which are strongly different and strongly the same, "portraits" of the differences between computing programs can be observed (See tables 16, 17, and 18.).

(2) Grouping of means responses for those topics that received "high" knowledge exit level ratings (mean response > 3.25) can be used for making "portraits" for each program (See Tables 13, 14, and 15.). The elements requiring high levels of knowledge for each discipline are really quite different.

These two arguments establish the validity of the hypothesis, that, there are very significant similarities and differences between the three computing disciplines.

(3) However significance testing also reveals that there are very definitely topics which are common to all three programs. Tables 19 through 22 are "portraits" of insignificantly different topics.

These three observations clearly validate the general hypothesis that there is a core of similar expectations as well as important differences for the three programs. Further observation confirms that topic content of each of the computing programs does indeed conform to the program "emphasis" statements contained in the hypotheses.

The implications of the results on future curriculum development efforts can be understood by examining Tables 13, 14, and 15, which are lists of top exit levels identified for each of the computing programs. "Use" and "Application" knowledge exit levels require considerable curriculum effort, whereas

"Awareness" and "Literacy" knowledge can be more easily obtained. If the significant differences in the highest exit level topics are correct, then it becomes evident that separate courses will be needed for the three programs.

7. CONCLUSIONS

1. Computer science, information systems, and software engineering professionals recognize and expect their students to learn many similar aspects of a body of computing knowledge.
2. Also, these professionals are able to express the depth of knowledge expected from program graduates.
3. Computer science and software engineering have largely similar expectations for their graduates, except for a software engineering focus on software development. This implies that software engineering is a reasonable subset of computer science at the undergraduate level.
4. Overall, software engineering looks a lot like computer science; the differences are small. But, unlike computer science, software engineering has its focus on software systems development.
5. One implication of the above conclusion is that a software engineering suite added to a curriculum for computer science might well be adequate to satisfy four-year undergraduate requirements for software engineering.
6. Exit expectations of information systems and computer science are very significantly different over most of the elements of the computing body of knowledge, while the two programs share all elements.
7. Information systems differs from software engineering similarly to how it differs from computer science, because computer science and software engineering are essentially the same.
8. The most significant topics (those with high knowledge exit level requirements) suggest a requirement for project courses that span several years (for all three computing programs), since high exit levels require several years to acquire.
9. The fact that there are many elements of the body of computing knowledge supports the notion there can be a core curriculum for the three computing programs.

10. **Using exit levels of knowledge is a useful way of distinguishing between programs.**

11. **Computer science, information systems, and software engineering professionals do accept a common body of computing knowledge.**

REFERENCE LIST

REFERENCE LIST

ACM 1968. ACM Curriculum Committee on Computer Science 1968. "Curriculum 68: Recommendations for the Undergraduate Program in Computer Science." Communications of the ACM, 11:3, March 1968, pp. 151 - 197.

ACM 1978. ACM Curriculum Committee on Computer Science 1978. "Curriculum 78: Recommendations for the Undergraduate Program in Computer Science. Communications of the ACM, 22:3, March 1979, pp. 147-166.

Ashenhurst, R. L. (ed.) 1972. "Curriculum Recommendations for Graduate Professional Programs in Information Systems" in ACM Committee on Computer Curricula of ACM Education Board. "ACM Recommendations for Information Systems, Volume II," ACM, NY, New York, 1983.

Berry, Daniel M. 1992. "Academic Legitimacy of the Software Engineering Discipline," Technical Report SEI-92-TR-034, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, Pa., November 1992.

BCS 1989. "A Report on Undergraduate Curricula for Software Engineering," The British Computer Society and The Institution of Electrical Engineers, June 1989.

Bloom, Benjamin S. (ed.) 1956. The Taxonomy of Educational Objectives: Classification of Educational Goals. Handbook 1: The Cognitive Domain. McKay Press, New York, NY, 1956.

Couger, J. (ed.) 1973. "Curriculum Recommendations for Undergraduate Programs in Information Systems," Communications of the ACM, 16:12, December 1973, pp. 727-749.

Davis, Gordon B., John T. Gorgone, J. Daniel Couger, David L. Feinstein, Herbert E. Longenecker, Jr. 1997. IS'97 Model Curriculum and Guidelines for Undergraduate Degree Programs in Information Systems, ACM, New York, NY and AITP (Formerly DPMA), Park Ridge, IL, 1997.

DPMA 1981. "DPMA Model Curriculum, 1981," DPMA, Park Ridge, IL.

DPMA 1986. "DPMA Model Curriculum, 1986," DPMA, Park Ridge, IL.

Ford, Gary, Norman Gibbs and James Tomayko 1987. "Software Engineering Education: An Interim Report from the Software Engineering Institute," Technical Report SEI-87-TR-8, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, Pa., May 1987.

Ford, Gary and Mark A. Ardis 1989. "1989 SEI Report on Graduate Software Engineering Education," Technical Report SEI-89-TR-21, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, Pa., June, 1989

Ford, Gary and Norman E. Gibbs 1989. "A Master of Software Engineering Curriculum: Recommendations from the Software Engineering Institute," IEEE Computer, September 1989, pp. 59-71.

Ford, Gary 1990. "1990 SEI Report on Undergraduate Software Engineering Education," Technical Report SEI-90-TR-3, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, Pa., March 1990.

Ford, Gary. 1991. "1991 SEI Report on Graduate Software Engineering Education," Technical Report SEI-91-TR-2, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, Pa., April 1991.

Ford, Gary 1994. "A Progress Report on Undergraduate Software Engineering Education," Technical Report SEI-94-TR-011, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, Pa., May 1994.

Freeman, P., A. I. Wasserman, and R. E. Fairley 1976. "Essential Elements of Software Engineering Education," Proceedings of the Second International Conference on Software Engineering, 1976, pp. 116-122.

Freeman, P. and A. I. Wasserman 1978. "A Proposed Curriculum for Software Engineering Education" Proceedings of the Third International Conference on Software Engineering, 1978, pp. 52-62.

Freeman, Peter 1987. "Essential Elements of Software Engineering Education Revisited," Software Engineering Education: The Educational Needs of the Software Community, Norman E. Gibbs and Richard E. Fairley, eds., Springer-Verlag, New York, New York, 1987, pp. 61-74.

Gibbs, Norman E. and Gary A. Ford 1986. "The Challenges of Educating the Next Generation of Software Engineers," Software Engineering Institute, Technical Report SEI-86-TM-7, Carnegie Mellon University, Pittsburgh, Pa. June 1986.

Gibbs, Norman E. 1989. "The SEI Education Program: The Challenge of Teaching Future Software Engineers," Communications of the ACM, 32:5, May 1989, pp. 594-605.

Glass, Robert L. 1992. "A comparative Analysis of the Topic Areas of Computer Science, Software Engineering, and Information Systems," Journal Systems Software, (19:4), pp. 277-289.

IEEE 1980. "Draft Report on MSE-80: A Graduate Program in Software Engineering," IEEE Software Engineering Subcommittee of the Computing Society Education Committee, 1980.

LaMalva, Patrick M. and George D. Peterson, 1996. "Accredited Programs in Computing," Communications of the ACM, 39:5, pp. 122-127.

Leventhal, Laura M. and Barbee T. Mynatt 1987. "Components of Typical Undergraduate Software Engineering Courses: Results from a Survey," IEEE Transactions on Software Engineering, Volume SE-13, Number 11, November 1987, pp. 1193-1198.

Longenecker, Herbert E., Jr. and David L. Feinstein (eds) 1991. "IS'90: The DPMA Model Curriculum for Information Systems for a Four Year Undergraduate Degree," DPMA, Park Ridge, IL, 1991.

Longenecker, Herbert E., Jr., David L. Feinstein, John T. Gorgone, Gordon B. Davis, J. Daniel Couger 1995. IS'95 Model Curriculum and Guidelines for Undergraduate Degree Programs in Information Systems (Draft Report), DPMA, ACM, AIS, 1995.

Longenecker, Herbert E., Jr., David L. Feinstein, John T. Gorgone, Gordon B. Davis, J. Daniel Couger 1996. IS'96 Model Curriculum and Guidelines for Undergraduate Degree Programs in Information Systems (Draft Report), DPMA, ACM, AIS, 1996.

NSF 1993. "A Framework for Academic Program in Informatics: Task Force Report," NSF Informatics Task Force, Center for Advanced Computer Studies, University of Southwestern Louisiana, Lafayette, La., November 1993.

Nunamaker, Jay F., J. Daniel Couger, and Gordon B. Davis 1982. "Information Systems Curriculum Recommendations for the 80s: Undergraduate and Graduate Programs," Communications of the ACM, 25:11, November 1982.

Parnas, David L. 1990. "Education for Computing Professionals," IEEE Computer 23, 1, January 1990, pp. 17-22.

Shaw, Mary 1986. "Education for the Future of Software Engineering," Technical Report SEI-86-TM-5, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, Pa., May 1986.

Shaw, Mary 1990. "Prospects for an Engineering Discipline of Software," Technical Report SEI-90-TR-20, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, Pa., September 1990.

SEI 1991. "Software Engineering Education Directory," Technical Report SEI-91-TR-9, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, Pa. May 1991.

Tomayko, James E. and Mary Shaw 1991. "Models for Undergraduate Project Courses in Software Engineering," Technical Report SEI-91-TR-10, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, Pa., August 1991.

Turner, A. Joe, and Allen Tucker (eds) 1991 "Computing Curricula 1991: Report of the ACM/IEEE-CS Joint Curriculum Task Force, Communications of the ACM, 34: 6, pp. 68-84, July 1991.

Washington Post, Employment Section, Washington, D.C., pp. K21-K61, September 17, 1997.

Wasserman, Anthony I. 1976. "Toward Improved Software Engineering Education," Software Engineering Education: Proceedings of an Interface Workshop, Anthony I. Wasserman and Peter Freeman, eds., Springer-Verlag, New York, New York, 1976, pp. 108-115.

GENERAL REFERENCES

Andersen, R., R. Conradi, J. Krogstie, G. Sindre, and A. Solvberg 1994. "Project Courses at the NTH: 20 Years of Experience," Software Engineering Education: 7th SEI CSEE Conference, Jorge L. Diaz-Herrera, ed., Springer-Verlag, New York, New York, 1994, pp. 177-188.

Ammann, Paul, Hassan Gomaa, Jeff Offutt, David Rine and Bo Sanden 1994. "A Five Year Perspective on Software Engineering Graduate Programs at George Mason University," Software Engineering Education: 7th SEI CSEE Conference, Jorge L. Diaz-Herrera, ed., Springer-Verlag, New York, New York, 1994, pp. 473-496.

Bailes, Paul A., Eric J. Salzman and Andreas Rosel 1989. "A Proposal for a Bachelor's Degree Program in Software Engineering," Software Engineering Education: SEI Conference 1989, Norman E. Gibbs, ed., Springer-Verlag, New York, New York, 1989, pp. 90-108.

Banville, Claude and Maurice Landry 1989. "Can the Field of MIS be Disciplined?," Communications of the ACM, 32: 1, January 1989, pp. 48-60.

Calliss, Frank W. and Debra Trantina Calliss 1994. "Suggested Scenarios of Software Maintenance Education," Software Engineering Education: 7th SEI CSEE Conference, Jorge L. Diaz-Herrera, ed., Springer-Verlag, New York, New York, 1994, pp. 329-340.

Coyle, Frank P., Dennis J. Frailey, Edward Forest, and Murat M. Tanik 1994. "Meeting the Needs of Industry: SMU's Master's Degree Program in Software Engineering," Software Engineering Education: 7th SEI CSEE Conference, Jorge L. Diaz-Herrera, ed., Springer-Verlag, New York, New York, 1994, pp. 543-554.

Deimel, Lionel E. 1989. "Programming and It's Relation to Computer Science Education and Software Engineering Education," Software Engineering Education: SEI Conference 1989, Norman E. Gibbs, ed., Springer-Verlag, New York, New York, 1989, pp. 253-256.

Denning, Peter J. 1984. "Educational Ruminations," Communications of the ACM, 27:10, October 1984, pp. 979-983.

Denning, Peter J. 1994. "Designing a Discipline of Software Design (Keynote Address)," Software Engineering Education: 7th SEI CSEE Conference, Jorge L. Diaz-Herrera, ed., Springer-Verlag, New York, New York, 1994

Epp, Edward C. 1994. "Project Oriented Courses," Software Engineering Education: 7th SEI CSEE Conference, Jorge L. Diaz-Herrera, ed., Springer-Verlag, New York, New York, 1994, pp. 165-178.

Freeman, Peter and Anthony I. Wasserman 1976. "Software Engineering Education: Workshop Overview," Software Engineering Education: Proceedings of an Interface Workshop, Anthony I. Wasserman and Peter Freeman, eds., Springer-Verlag, New York, New York, 1976, pp. 2-4.

Gibbs, Norman E. and Allen B. Tucker 1986. "A Model Curriculum for a Liberal Arts Degree in Computer Science," Communications of the ACM, 29: 3, March 1986, pp. 202-210.

Glass, Robert L. 1987. "Software Engineering and Computer Science: How Do They Differ?," Software Engineering Education: The Educational Needs of the Software Community, Norman E. Gibbs and Richard E. Fairley, eds., Springer-Verlag, New York, New York, 1987, pp. 75-77.

Hartvigsen, Gunnar 1994. "Reducing the Gap Between Academic Theory and Professional Practice in Software Engineering Education," Software Engineering Education: 7th SEI CSEE Conference, Jorge L. Diaz-Herrera, ed., Springer-Verlag, New York, New York, 1994, pp. 263-273.

Habermann, A. Nico 1987. "The Environment for the Software Engineer," Software Engineering Education: The Educational Needs of the Software Community, Norman E. Gibbs and Richard E. Fairley, eds., Springer-Verlag, New York, New York, 1987, pp. 78-86.

Humphrey, Watts S. 1990. Managing the Software Process, Addison-Wesley Publishing Company, Reading Ma., 1990.

Johnston, Lorraine and Philip Dart 1994. "Building on Experience: An Undergraduate Course with Two Year-Long Projects," Software Engineering Education: 7th SEI CSEE Conference, Jorge L. Diaz-Herrera, ed., Springer-Verlag, New York, New York, 1994, pp. 345-357.

Lewis, Philip M. 1989. "Information Systems is an Engineering Discipline," Communications of the ACM, 32: 9, pp. 1045 - 1047, September 1989.

Musa, John D. 1985. "Software Engineering: The Future of a Profession," IEEE Software, Volume 2, Number 1, January 1985, pp. 55-62.

Modesitt, Kenneth L. 1994. "When the Golden Arches Gang Aft Agley: Incorporating Software Engineering into Computer Science," Software Engineering Education: 7th SEI CSEE Conference, Jorge L. Diaz-Herrera, ed., Springer-Verlag, New York, New York, 1994, pp. 35-61.

Nash, John C. and Mary M. Nash 1989. "Information Systems is a General Discipline," Communications of the ACM, 32:12, December 1989, p 1395.

Pfleeger, Shari Lawrence 1987. Software Engineering, The Production of Quality Software, Macmillan Publishing Company, New York, NY, 1987, 443 pp.

Sodhi, Jag 1991. Software Engineering Methods, Management, and CASE Tools, TAB Professional and Reference Books, Blue Ridge Summit, Pa., 1991, 383 pp.

Steward, Donald V. 1987. Software Engineering with Systems Analysis and Design, Brooks/Cole Publishing Company, Monterey, Calif., 1987, 414 pp.

Van Scoy, Frances L. 1989. "Developing an Undergraduate Software Engineering Curriculum within an Existing Computer Science Program," Software Engineering Education: SEI Conference 1989, Norman E. Gibbs, ed., Springer-Verlag, New York, New York, 1989, pp. 294-303.

Weis, Richard Louis 1989. "Software Engineering in a BS in Computer Science," Software Engineering Education: SEI Conference 1989, Norman E. Gibbs, ed., Springer-Verlag, New York, New York, 1989, pp. 304-309.

Werth, Laurie Honour 1994. "An Adventure in Software Process Improvement," Software Engineering Education: 7th SEI CSEE Conference, Jorge L. Diaz-Herrera, ed., Springer-Verlag, New York, New York, 1994, pp. 191-210.

Zweben, Stuart H. 1989. "Integrating Software Engineering into an Undergraduate Computer Science Curriculum," Software Engineering Education: SEI Conference 1989, Norman E. Gibbs, ed., Springer-Verlag, New York, New York, 1989, pp. 310-313.

APPENDICES

APPENDIX A
COMMON BODY OF COMPUTING KNOWLEDGE

The Common Body of Computing Knowledge was included in IS'97 (Davis et al).

1.0 Information technology
 1.1 Computer architectures
  1.1.1 Fundamental data representation: non-numeric, numeric (integers, reals, errors, precision)
   1.1.1.1 Basic machine representation of numeric data
   1.1.1.2 Basic machine representation of non-numeric data
   1.1.1.3 Finite precision of integer and floating point number representation
   1.1.1.4 Errors in computer arithmetic and related portability issues
   1.1.1.5 Basic concepts of computer architecture
  1.1.2 Physical representation of digitized information: e.g., data, text, image, voice, video
  1.1.3 CPU architectures: CPU, memory, registers, addressing modes, instruction sets
   1.1.3.1 Basic organization: von Neumann, block diagram, datapaths, control path, functional units, instruction cycles
   1.1.3.2 Instructions and addressing modes: instruction sets and types
   1.1.3.3 Instructions and addressing modes: assembly-machine language
   1.1.3.4 Addressing modes
   1.1.3.5 Control unit; instruction fetch and execution, operand fetch
   1.1.3.6 CISC, RISC
   1.1.3.7 Computer organization
   1.1.3.8 Memory systems
  1.1.4 Computer system components: busses, controllers, storage systems, peripheral devices
   1.1.4.1 Peripherals: I/O and interrupts
   1.1.4.2 Peripherals: input/output control methods, interrupts
   1.1.4.3 Peripherals: external storage, physical organization and drives
   1.1.4.4 Auxiliary storage, tape, optical
   1.1.4.5 Storage systems and technology
   1.1.4.6 Space allocation, hierarchy
   1.1.4.7 Main memory organization, bus operations, cycle times for selection and addressing
   1.1.4.8 Cache memory, read/write
   1.1.4.9 Virtual memory
   1.1.4.10 Interfaces between computers and other devices (sensors, effectors, etc.)
  1.1.5 Multiprocessor architectures
   1.1.5.1 Systems architectures (single multi-processing and distributed processing, stack, array, vector, multiprocessor and hypercube architectures, supercomputers)
   1.1.5.2 Client server technologies
  1.1.6 Digital logic and systems
   1.1.6.1 Logic elements and switching theory; minimization concepts and implementation of functions
   1.1.6.2 Propagation delays and hazards

1.1.6.3 Demultiplexers, multiplexers, decoders, encoders, adders, subtractors, comparators, shift registers, counters
1.1.6.4 ROM, PROM, EPROM, EAPROM, RAM
1.1.6.5 Analysis and synthesis of synchronous circuits, asynchronous vs synchronous circuits
1.1.6.6 Register transfer notation, conditional and unconditional
1.1.6.7 Algorithmic state machines, steering networks, load transfer signals
1.1.6.8 Tristates and bus structures
1.1.6.9 Block diagrams, timing diagrams, transfer language

1.2 Algorithms and data structures

1.2.1 Formal problems and problem solving
1.2.1.1 Problem solving strategies using greedy algorithms
1.2.1.2 Problem solving strategies using divide and conquer algorithms
1.2.1.3 Problem solving strategies using back- tracking algorithms
1.2.1.4 Software design process; from specification to implementation
1.2.1.5 Problem recognition statement and algorithmic determination; procedural abstraction: parameters
1.2.1.6 Implementation strategies (top-down, bottom-up; teams vs individual; management task)
1.2.1.7 Formal verification concepts
1.2.1.8 Formal models of computation

1.2.2 Basic data structures: lists, arrays, strings, records, sets, linked-lists, stacks, queues, trees, graphs

1.2.3 Complex data structures: e.g. of data, text, voice, image, video, hypermedia

1.2.4 Abstract data types
1.2.4.1 Purpose and implementation of abstract data types
1.2.4.2 Informal specifications
1.2.4.3 Formal specifications, pre-conditions and post-conditions, algebraic specifications for abstract data types
1.2.4.4 Modules, cohesion, coupling; data flow diagrams, and conversion to hierarchy charts
1.2.4.5 Correctness, verification and validation: pre- and post-conditions, invariants, elementary proofs of code and design reading, structured walkthroughs
1.2.4.6 Control structures; selection, iteration, recursion; data types and their uses in problem solving

1.2.5 File structures: sequential, direct access, hashing, indexed
1.2.5.1 Files (structure, access methods): file layouts; fundamental file concepts; sequential files; non-sequential files
1.2.5.2 Files (structure, access methods): directories, contents and structure, naming, searching, access, backups
1.2.5.3 Files (structure, access methods): system security overview, security methods and devices, protection, access, authentication

1.2.6 Sorting and searching data structures and algorithms
1.2.6.1 Sorting algorithms (shell sort, bucket sort, radix sort, quick sort), editing, reporting, updating
1.2.6.2 Searching algorithms (serial search, binary search, and binary search tree)
1.2.6.3 Searching, hashing, collision resolution

1.2.7 Algorithm efficiency, complexity and metrics
1.2.7.1 Asymptotic analysis at upper and average bounds; big "O", little "O"
1.2.7.2 Time vs space trade-offs in algorithms
1.2.7.3 Complexity classes P, NP, P-space; tractable and intractable problems
1.2.7.4 Lower bound analysis (for sorting)
1.2.7.5 NP-completeness
1.2.7.6 O (n "squared") sorting algorithms

1.2.7.7 O (n log n) sorting algorithms
1.2.7.8 Backtracking, parsing, discrete simulations, etc.
1.2.7.9 Fundamentals of analysis of algorithms
1.2.8 Recursive algorithms
1.2.8.1 Recursive algorithms connection with mathematical induction
1.2.8.2 Comparison of iterative and recursive algorithms
1.2.9 Neural networks and genetic algorithms
1.2.10 Advanced considerations
1.2.10.1 Computable functions: models of computable functions selected from Turing machines, RAM, (partial) recursive functions, lambda calculus, Church's thesis
1.2.10.2 Machines, e.g. Universal Turing Machine
1.2.10.3 Decision problems: recursive and recursively enumerable problems; undecidable problems
1.2.10.4 Models of parallel architectures
1.2.10.5 Algorithms for parallel architectures
1.2.10.6 Mathematical problems: well-conditioned and ill-conditioned problems
1.2.10.7 Mathematical problems: iterative approximation to mathematical problems; Newton's method; Gaussian elimination
1.2.10.8 Mathematical problems: error classification; computational, representational, and methodological distinctions
1.2.10.9 Mathematical problems: applications of iterative approximation methods in sciences and engineering
1.2.10.10 Bounds on computing: computability and algorithmic intractability
1.3 Programming languages
1.3.1 Fundamental programming language structures; comparison of languages and applications
1.3.2 Machine and assembly level languages
1.3.3 Procedural languages
1.3.3.1 Procedural programming advantages and disadvantages
1.3.3.2 Basic type declarations; arithmetic operators and assignment; conditional statements; loops and recursion
1.3.3.3 Procedures, functions, and parameters; arrays and records
1.3.4 Non-procedural languages: logic, functional
1.3.5 Fourth-generation languages
1.3.6 Object oriented extensions to languages
1.3.7 Programming languages, design, implementation and comparison
1.3.7.1 History of early languages
1.3.7.2 Evolution of procedural languages
1.3.7.3 Evolution of non-procedural languages
1.3.7.4 Virtual computers
1.3.7.5 Elementary and structured data types
1.3.7.6 Creation and application of user defined data types
1.3.7.7 Expressions, order of evaluation, and side-effects
1.3.7.8 Subprograms and coroutines as abstractions of expressions and statements
1.3.7.9 Exception handling
1.3.7.10 Mechanisms for sharing and restricting access to data
1.3.7.11 Static vs dynamic scope, lifetimes, visibility
1.3.7.12 Parameter passing mechanisms; reference, value, name, result, etc.
1.3.7.13 Varieties of type checking disciplines and their mechanics
1.3.7.14 Stack-based application of storage
1.3.7.15 Heap-based application of storage
1.3.7.16 Finite state automata as restricted models of computation and acceptors of regular expressions

1.3.7.17 Application of regular expressions to programming language analysis
1.3.7.18 Use of context-free grammars as a formal description device for programming language syntax
1.3.7.19 Equivalence of context free grammar and pushdown automata
1.3.7.20 Use of pushdown automata in parsing programming languages
1.3.7.21 Language translation process, compilers to interpreters
1.3.7.22 Programming language semantics
1.3.7.23 Functional programming paradigms and languages
1.3.7.24 Parallel programming constructs
1.3.7.25 Procedural languages: implementation issues; performance improvement, debugging, anti-bugging
1.3.7.26 Compilers and translators
1.3.7.27 Very high level languages: SQL, 4th-GL
1.3.7.28 Object-oriented design, languages, and programming
1.3.7.29 Logic programming languages: LISP, PROLOG; logic oriented programming
1.3.7.30 Code generators
1.3.7.31 Expert system shells
1.3.7.32 Software design languages

1.4 Operating systems

1.4.1 Architecture, goals and structure of an operating system; structuring methods, layered models, object-server model
1.4.2 Interaction of operating system and hardware architecture
1.4.3 Process management: concurrent processes, synchronization
1.4.3.1 Tasks, processes, dispatching context switchers, role of interrupts
1.4.3.2 Structures, ready list, process control blocks
1.4.3.3 Concurrent process execution
1.4.3.4 Sharing access, race conditions
1.4.3.5 Deadlock; causes, conditions, prevention
1.4.3.6 Models and mechanisms (e.g., busy waiting, spin locks, Deker's algorithm, semaphores, mutex locks, regions, monitors
1.4.3.7 Preemptive and non-preemptive switching
1.4.3.8 Schedulers and scheduling policies
1.4.4 Memory management
1.4.4.1 Physical memory and registers
1.4.4.2 Overlays, swapping, partitions
1.4.4.3 Pages and segments
1.4.4.4 Placement and replacement policies
1.4.4.5 Thrashing, working sets
1.4.4.6 Free lists, layout; servers, interrupts; recovery from failures
1.4.4.7 Memory protection; recovery management
1.4.5 Resource allocation and scheduling
1.4.5.1 Protocol suites (communications and networking); streams and datagrams
1.4.5.2 Internetworking and routing; servers and services
1.4.5.3 Types of operating systems: single user, multi-user, network
1.4.5.4 Synchronization and timing in distributed and real time systems
1.4.5.5 Special concerns in real-time systems; failures, risks, and recovery
1.4.5.6 Operating system utilities
1.4.5.7 Hardware evolution; economic forces and constraints
1.4.5.8 Architecture of real-time and embedded systems
1.4.5.9 Special concerns in embedded real-time systems: hard-timing requirements; reliability, robustness, and fault tolerance; input and output considerations; awareness of issues

pertaining to time; concurrency; complex interfaces of device/device and device/software; inadequacy of testing for real-time systems

1.4.6 Secondary storage management

1.4.7 File and directory systems

1.4.8 Protection and security

1.4.9 Distributed operating systems

1.4.10 OS support for human interaction: e.g., GUI, interactive video

1.4.11 OS interoperability and compatibility: e.g., open systems

1.4.12 Operating system utilities, tools, commands and shell programming

1.4.13 System administration and management

1.4.13.1 System bootstrapping/initial program load

1.4.13.2 System generation

1.4.13.3 System configuration

1.4.13.4 Performance analysis, evaluation and monitoring

1.4.13.5 System optimization and tuning

1.4.13.6 System management functions: backup, security and protection, adding and deleting users

1.5 Telecommunications

1.5.1 International telecommunication standards, models, trends

1.5.1.1 Computer networks and control: topologies, common carriers, equipment configuration, error detection and correction, polling and contention protocols, security and encryption

1.5.1.2 Network design and management: network architectures (ISO, SNA, DNA), protocols (X.25, ISO, etc.)

1.5.2 Data transmission: media, signaling techniques, transmission impairments, encoding, error detection, compression

1.5.2.1 Communications system technology: transmission media, analog-digital, communications hardware and software

1.5.3 Line configuration: error control, flow control, multiplexing

1.5.4 Local area networks

1.5.4.1 Topologies, medium access control, multiplexing

1.5.4.2 Local area networks and WANs: topology, gateways, uses (functions and office automation), PBXs

1.5.4.3 Requirements determinations, performance monitoring and control, economics

1.5.4.4 Architecture of distributed systems

1.5.4.5 Hardware aspects of distributed systems

1.5.5 Wide area networks: switching techniques, broadcast techniques, routing

1.5.6 Network architectures and protocols

1.5.7 Internetworking

1.5.8 Network configuration, performance analysis and monitoring

1.5.9 Network security: encryption, digital signatures, authentication

1.5.10 High-speed networks: e.g., broadband ISDN, SMDS, ATM, FDDI

1.5.11 Emerging networks: ATM, ISDN, satellite nets, optic nets, etc., integrated voice, data and video

1.5.12 Application: e.g., client server, EDI, EFT, phone network, e-mail, multimedia, video conferencing, value-added networks

1.5.12.1 Methods of transmitting graphical and video information using telecomm, data compression, client-server display techniques, e.g., AOL interface, XWindows

1.6 Database

1.6.1 DBMS: features, functions, architecture

1.6.1.1 DBMS (features, functions, architecture); components of database system (data, dictionary, application programs, users, administration)

1.6.1.2 DBMS: overview of relational algebra
1.6.1.3 Logical design (DBMS independent design): ER, object oriented
1.6.2 Data models: relational, hierarchical, network, object, semantic object
1.6.2.1 Relational data model terminology; mapping conceptual schema to a relational schema
1.6.2.2 Conceptual modeling (e.g., entity-relationship, object-oriented)
1.6.3 Normalization
1.6.4 Integrity (referential, data item, intra-relation): representing relationships; entity and referential integrity
1.6.5 Data definition languages
1.6.6 Application interface
1.6.6.1 Function supported by typical database system: access methods, security, deadlock and concurrency problems, 4th generation environments
1.6.6.2 DML, query, QBE, SQL, etc.: database query language; data definition, query form, update sub-language, expressing constraints, referential integrity, embedding in a procedural language
1.6.6.3 Application and user interfaces (DML, query, QBE, SQL)
1.6.7 Intelligent query processors and query organization
1.6.8 Distributed databases
1.6.9 DBMS products: recent developments in database systems (e.g., hypertext, hypermedia, optical disks)
1.6.10 Database machines
1.6.11 Data and database administration
1.6.11.1 Data administration
1.6.11.2 Database administration: social impact of database systems; security and privacy
1.6.11.3 Ownership of data and application systems
1.6.12 Data dictionary, encyclopedia, repository
1.6.13. Information retrieval: e.g. image processing, hypermedia
1.7 Artificial intelligence
1.7.1 Knowledge representation
1.7.1.1 History, scope and limits of artificial intelligence: the Turing test
1.7.1.2 Social, ethical, legal, and philosophical aspects of artificial intelligence
1.7.1.3 Problems and state spaces
1.7.2 Knowledge engineering
1.7.3 Inference processing
1.7.3.1 Basic control strategies (e.g., depth-first, breadth-first)
1.7.3.2 Forward and backward reasoning
1.7.3.3 Heuristic search (e.g., generate & test, hill climb, breadth-first search, means-ends analysis, graph search, minimax search)
1.7.3.4 Expert systems and shells
1.7.4 Other techniques: fuzzy logic, CASE-based reasoning, natural language and speech recognition
1.7.5 Knowledge-based systems
1.7.5.1 Natural language, speech and vision
1.7.5.2 Pattern recognition
1.7.5.3 Machine learning
1.7.5.4 Robotics
1.7.5.5 Neural networks
2.0 Organizational and management concepts
2.1 General organization theory
2.1.1 Hierarchical and flow models of organizations
2.1.2 Organizational work groups
2.1.3 Organizational span: single user, work group, team, enterprise, global

2.1.4 Role of IS within the enterprise: strategic, tactical and operations
2.1.5 Effect of IS on organizational structure; IS and continuous improvement
2.1.6 Organizational structure: centralized, decentralized, matrix
2.1.7 Organizational issues pertaining to use of software systems in organizations
2.2 Information systems management
2.2.1 IS planning
2.2.1.1 Alignment of IS planning with enterprise planning
2.2.1.2 Strategic IS planning
2.2.1.3 Short-range IS planning
2.2.1.4 Re-engineering
2.2.1.5 Continuous improvement
2.2.2 Control of the IS function: e.g., EDP auditing, outsourcing
2.2.3 Staffing and human resource management
2.2.3.1 Skills planning
2.2.3.2 Staff performance management
2.3.3.3 Empowerment/job ownership
2.2.3.4 Education and training
2.2.3.5 Competition, cooperation and reward structures
2.2.4 IS functional structures -- internal vs outsourcing
2.2.5 Determining goals and objectives of the IS organization
2.2.6 Managing IS as a business: e.g., customer definition, defining IS mission, IS critical success factors
2.2.7 CIO and staff functions
2.2.8 IS as a service function: performance evaluation -- external/internal, marketing of services
2.2.9 Financial administration of IS: e.g., funding and chargeout
2.2.10 Strategic use of IS: e.g., competitive advantage and IS, process re-engineering, IS and quality, IS global impact and international considerations
2.2.11 End user computing support, role and functions
2.2.12 IS policy and operating procedures formulation and communication
2.2.13 Backup, disaster planning and recovery
2.2.14 Management of emerging technologies
2.2.15 Management of sub-functions
2.2.15.1 Telecommunications management
2.2.15.2 Computer facilities management: e.g., automated operations of distributed processing, capacity planning, site maintenance
2.2.15.3 Management of group decision support systems
2.2.15.4 Data administration
2.2.15.5 Ownership of data and application systems
2.2.15.6 Optimizing the climate for creativity
2.2.15.7 Quality management: e.g., reliability and quality engineering; QC teams
2.2.15.8 Management consulting relationships, outsourcing
2.2.15.9 Managing for resource contention
2.2.15.10 Operational issues associated with system installation, transition, operation, and retirement
2.2.15.11 Controlling activities and disciplines which support software evolution and maintenance
2.2.15.12 Software engineering activities: development, control, management, operations
2.2.16 Security and control, viruses and systems integrity
2.2.17 Computer operations management: e.g. tape/DASD management, scheduling, automation-cross functional context
2.3 Decision theory
2.3.1 Measurement and modeling

2.3.2 Decisions under certainty, uncertainty, risk
2.3.3 Cost/Value of information, competitive value of IS
2.3.4 Decision models and IS: optimizing, satisficing
2.3.5 Group decision process
2.4 Organizational behavior
2.4.1 Job design theory
2.4.3 Group dynamics
2.4.4 Teamwork, leadership and empowerment
2.4.5 Use of influence, power and politics
2.4.6 Cognitive styles
2.4.7 Negotiating and negotiating styles
2.4.8 Consensus building
2.7 Managing the process of change
2.7.1 Reasons for resistance to change
2.7.2 Strategies for motivating change
2.7.3 Planning for change
2.7.4 Managing change
2.8 Legal and ethical aspects of IS
2.8.1 Software sales, licensing, and agency
2.8.2 Contract fundamentals
2.8.2.1 Contract law
2.8.3 Privacy law
2.8.4 Agencies and regulatory bodies
2.8.5 Protection of intellectual property rights
2.8.5.1 Protection of intellectual property
2.8.5.2 Forms of intellectual property, means for protecting it, and penalties for violating it
2.8.5.3 Ethics (plagiarism, honesty, privacy): uses, misuses, and limits of computer technology
2.8.6 Ethics: plagiarism, honesty, codes of ethics
2.8.6.1 Ethics: plagiarism, honesty, privacy
2.8.6.2 Ethics: Social and ethical responsibilities of the computing professional
2.8.7 Risks, losses and liability in computing applications
2.8.8 Warranties
2.9 Professionalism
2.9.1 Current literature periodicals, professional, academic journals
2.9.2 Certification issues
2.9.3 Professional organizations: e.g. DPMA, ACM, TIMS, ASM, DSI, ACE, IEEE, ASQC, AIS, IAIM, INFORMS
2.9.4 Professional conferences
2.9.6 IS industry: manufacturers, OEMs, system integrators, software developers
2.9.7 Historical and social context of computing
2.10 Interpersonal Skills
2.10.1 Communication skills
2.10.2 Interviewing, questioning and listening
2.10.3 Presentation skills
2.10.3.1 Oral and written communications
2.10.3.2 Graphics and use of multimedia
2.10.3.3 Training: goals, objectives, computer based
2.10.4 Consulting skills
2.10.5 Writing skills
2.10.5.1 Fundamentals of technical writing
2.10.5.2 Principles and standards for documentation
2.10.5.3 Development of software documentation

2.10.5.4 Documentation tools
2.10.6 Proactive attitude and approach
2.10.7 Personal goal setting, decision making, and time management
2.10.8 Principle centered leadership
2.10.9 Principles of negotiation
3.0 Theory and development of systems
3.1 Systems and information concepts
3.1.1 General systems theory
3.1.2 Systems concepts: e.g., structure, boundaries, states, objectives
3.1.2.1 Fundamental concepts of information theory
3.1.2.2 Reasoning about organizational systems, software products and processes
3.1.2.3 Relationships of users and suppliers to the system
3.1.3 Properties of open systems
3.1.4 System components and relationships
3.1.5 Systems control: standards, control theory, feedback, loops, measurement, quality
3.1.6 Properties of information systems
3.2 Approaches to systems development
3.2.1 Systems development models: e.g., SDLC, prototyping
3.2.1.1 Systems development life cycle: software life-cycle models (iterative enhancement, phased development, spiral, waterfall)
3.2.1.2 Developing with prototyping
3.2.1.3 Developing with packages
3.2.1.4 Data oriented development techniques
3.2.1.5 Process oriented development techniques
3.2.1.6 Object oriented development techniques: bottom-up design; support for reuse
3.2.1.7 Systems engineering considerations
3.2.1.8 Software as a component of a system
3.2.1.9 Software process and product-life cycle models
3.2.1.10 Software generation methods and tools: design and coding from scratch, program and application generators, very high level languages, reusable components
3.2.1.11 System design methods and tools
3.2.2 Package acquisition and implementation
3.2.3 Integrating software components
3.2.4 User developed systems
3.2.5 Selecting a systems development approach
3.3 Systems development concepts and methodologies
3.3.1 Organizational and software process modeling
3.3.1.1 Modeling concepts
3.3.1.2 Advanced modeling concepts, including asynchronous and parallel models
3.3.2 Data modeling: e.g., entity-relationship diagrams, normalization
3.3.3 Data oriented methodologies
3.3.4 Process oriented methodologies
3.3.5 Behavior oriented (event modelling) methodologies
3.3.6 Object oriented methodologies
3.3.7 Software engineering process and products
3.4 Systems development tools and techniques
3.4.1 CASE
3.4.1.1 Methodologies (information engineering, Jackson Techniques, Yourdon, C. F. Martin, etc.): software design objectives
3.4.1.2 Tools: CASE tools, code generators, GDSS
3.4.1.3 Tools (CASE tools, code generators, GDSS): specification and design tools; implementation tools

3.4.2 Group-based methods: e.g., JAD, structured walkthroughs, design and code reviews
3.4.3 Software implementation concepts and tools: e.g., data dictionary, repository, application generator, reuse, program generators, software implementation languages

3.5 Application planning
3.5.1 Infrastructure planning: hardware, communications, database, site
3.5.2 Planning the IS architecture
3.5.3 Planning for operations
3.5.4 Metrics for size, function points, control of complexity
3.5.5 Planning for IS security, privacy and control

3.6 Risk management
3.6.1 Feasibility assessment
3.6.2 Risk management principles
3.6.3 Contingency planning

3.7 Project management
3.7.1 Project planning and selection of appropriate process model; project scheduling and milestones
3.7.2 Project organization, management, principles, concept and issues
3.7.2.1 Project management organizational issues
3.7.2.2 Project management principles, concepts and issues
3.7.3 Work breakdown structures and scheduling
3.7.4 Project staffing considerations: e.g., matrix management, human factors, team organization, reporting
3.7.5 Project control: planning, cost estimation, resource allocation, software technical reviews, measurement, analysis, feedback, communications, ensuring quality, scheduling, milestones
3.7.5.1 Project management documentation
3.7.5.2 Representations of project scheduling
3.7.5.3 Project economics: cost estimation techniques and tools: cost/benefit analysis: risk analysis; etc.
3.7.5.4 Project scheduling tools
3.7.6 Managing multiple projects
3.7.7 Management concerns; stress and time management
3.7.8 Systems documentation
3.7.9 User documentation (e.g., reference manuals, operating procedures, on-line documentation)
3.7.10 System metrics
3.7.11 Scoping and scope control
3.7.12 Configuration management
3.7.12.1 Principles and concepts of configuration management
3.7.12.2 Role in controlling system evolution
3.7.12.3 Role in maintaining product integrity
3.7.12.4 Documentation: change controls, version controls, etc.
3.7.12.5 Organizational structures for configuration management
3.7.12.6 Configuration management plans
3.7.12.7 Configuration management tools
3.7.13 System development quality assurance
3.7.14 Project tracking: e.g., PERT, Gantt
3.7.15 Project close-down

3.8 Information and business analysis
3.8.1 Problem opportunity identification: e.g., service requests, from planning process
3.8.2 Relating the application to the enterprise model
3.8.3 Requirements determination and specification

3.9 Information systems design
3.9.1 Design: logical, physical

3.9.1.1 System design methods and tools
3.9.1.2 Role of software design versus system design
3.9.1.3 Hardware-software tradeoffs for system performance and flexibility
3.9.1.4 Design of high-level interfaces, hardware to software and software to software
3.9.1.5 System performance prediction
3.9.1.6 System modeling techniques and representations
3.9.1.7 Object oriented system design technique
3.9.1.8 System design techniques: iterative design technique, modeling, etc.
3.9.1.9 System design flexibility

3.9.2 Design methodologies: e.g., real time, object oriented, structured
3.9.3 Design objectives: e.g., usability, performance
3.9.4 Techniques to enhance the creative design process
3.9.5 Information presentation alternatives: cognitive styles
3.9.6 Human-computer interaction (e.g., ergonomics, graphical-user interfaces, voice, touch)

3.9.6.1 User interfaces (voice, touch...)
3.9.6.2 Ergonomics
3.9.6.3 Common user access
3.9.6.4 User interfaces; menu systems, command languages, direct manipulation, common interface tool kits
3.9.6.5 Graphics output devices and their properties
3.9.6.6 Graphics primitives and their properties
3.9.6.7 Graphics software systems; general graphics standards
3.9.6.8 Architecture of window managers and user interfaces
3.9.6.9 Architecture of toolboxes and programming support environments
3.9.6.10 Representation of graphic data and sound
3.9.6.11 Design techniques for human-computer interface problems: device independence, virtual terminals, etc.
3.9.6.12 Human factors associated with human-computer interfaces: assumptions about class of user, handling input errors, screen design, etc.

3.9.7 Software development

3.9.7.1 Software requirements: principles; types (functional, performance and other); analysis; identification techniques (prototyping, modeling, simulation); communication with customer; tools
39.7.2 Software specifications: objectives; standards; types (functional, performance, reliability, other); formal models; representations; documents (standards, structure, content, users, completeness, consistency); techniques; specification of quality attributes; formal specification languages and tools
3.9.7.3 Software design: principles of design (abstraction, information hiding, modularity, reuse, prototyping); paradigms for well-understood systems; levels of design; documentation; representations of designs; design of sub-systems; assessment of design quality; languages and tools; methods, practices and techniques
3.9.7.4 Software quality assurance: issues, definitions, standards, quality assurance as a controlling discipline, factors affecting quality, quality concerns in phases of the SDLC, metrics, organizational structures for quality assurance, plans, documentation, quality assurance project teams, quality and security, industrial practice
3.9.7.5 Software correctness and reliability: principles, concepts, modeling, methods
3.9.7.6 Verification and validation of software quality assurance: role and methods, formal models, independent verification and validation teams, tools, reports
3.9.7.7 Software implementation: relationship of software design to implementation; relationship of programming support environments to software implementation process; relationship of implementation language to design principles; tools; assessment (coding

standards, metrics, etc.); other implementation considerations and issues (language structures and programming techniques, reuse, application generators, etc.)

3.9.7.8 Software and hardware system integration: methods, plans, tests (including incremental testing during development), assessment and documentation of test results, diagnosing system faults, simulation of missing hardware

3.9.7.9 Software testing: role, principles and standards; relationship of quality assurance to testing; methods; levels of testing (unit, system, integration, acceptance, etc.); plans; audits; limitations; statistical methods; formal models; documentation; tools: test and evaluation teams; building test environments; test case generation; regression testing; black-box/white-box testing; technical reviews; performance analysis; results analysis and reports

3.10 Systems implementation and testing strategies

3.10.1 Systems construction

3.10.2 Software systems construction: e.g., programming, unit testing, load module packaging

3.10.3 Software integration: e.g., packages

3.10.4 Systems conversion: approaches, planning, implementation

3.10.5 Systems integration and system testing: verification and validation, test plan generation, testing (acceptance testing, unit testing, integration testing, regression testing)

3.10.6 Training: e.g., user, management, operation, systems, training materials

3.10.7 Software project management: scoping, scheduling, configuration management, quality assurance; software reliability issues (safety, responsibility, risk assessment); maintenance

3.10.8 Systems installation

3.10.9 Post implementation review

3.11 Systems operation and maintenance

3.11.1 Service request and change control

3.11.2 Reverse and re-engineering

3.11.3 Tuning and balancing

3.11.4 Systems and software maintenance concepts

3.11.4.1 Kinds of software maintenance: perceptive, adaptive, corrective

3.11.4.2 Designing software for maintainability

3.11.4.3 Software maintenance techniques: program reading, reverse engineering, etc.

3.11.4.4 Software maintenance models

3.12 Systems development for specific types of information systems

3.12.1 Transaction processing systems

3.12.2 Management information systems

3.12.3 Group support systems

3.12.4 Decision support systems/expert systems

3.12.5 Executive support systems

3.12.6 Office systems

3.12.7 Collaborative systems

3.12.8 Work-flow systems

3.12.9 Functional support systems: e.g., process control, marketing

3.12.10 Interorganizational systems

APPENDIX B
SUMMARY BODY OF KNOWLEDGE TOPICS
(Aggregated under level-2 of the common body of computing knowledge)

1.0. Information Technology (Group I)

1.1. Computer Architectures

TOPIC

1. **Fundamental data presentation and physical representation of digitized information -- numeric, non-numeric (integers, reals, errors, precision); data, text, image, voice, video**

2. **CPU architectures and computer system components -- CPU, memory, registers, addressing modes, instruction sets; buses, controllers, storage systems, peripheral devices**

3. **Multiprocessor architectures -- single multiprocessing and distributed processing, stack, array, vector, multiprocessor and hypercube architectures, super computers, client server technologies**

4. **Digital logic and systems -- logic elements and switching theory, propagation delays and hazards, register transfer notation, block diagrams, timing diagrams, multiplexers, etc.**

1.2. Algorithms and Data Structures

5. **Formal problems and problem solving -- problem solving, design, and implementation strategies for algorithms and data structures; formal verification concepts; formal models; etc.**

6. **Basic data structures -- lists, arrays, strings, records, sets, linked-lists, stacks, queues, etc.**

7. **Complex data structures -- data, text, voice, image, video , hypermedia, etc.**

8. **Abstract data types -- purpose and implementation, specifications, modules, cohesion, coupling, correctness, verification, validation, invariants, proofs, control structures, etc.**

9. **File structures and access methods -- sequential, direct access, hashing, indexed; file layouts: directories; contents and structure; access; backups; system security overview; etc.**

10. **Sorting and searching data structures and algorithms -- quick sort, binary search, etc.**

11. **Algorithm efficiency, complexity and metrics -- analysis, big "O", little "O", time vs. space trade-offs, NP- completeness, backtracking, parsing, discrete simulations, etc.**

12. **Recursive algorithms**

13. Advanced consideration of algorithms -- neural networks and genetic algorithms, algorithms for parallel architectures, mathematical problems, bounds on computing, etc.

1.3. Programming Languages

14. Programming languages -- fundamental programming language structures; procedural and non-procedural languages; 4th-GL; object oriented extensions to languages; etc.

15. Machine and assembly languages

16. Design, implementation, and comparison of programming languages -- history, evolution; language translation process; software design languages, compilers and translators; etc.

1.4. Operating Systems

17. Architecture, goals and structure of operating systems -- structuring methods, layered models, object server model

18. Interaction of operating system and hardware architecture

19. Operating system process management -- concurrent processes; synchronization; tasks; deadlock; semaphores preemptive and non-preemptive switching; schedulers; etc.

20. Memory management -- physical memory and registers; overlays, swapping, partitions; pages and segments; thrashing; interrupts; memory protection; recovery management; etc.

21. Resource allocation and scheduling -- architecture of real-time and embedded systems; reliability, robustness and fault tolerance; operating system utilities; complex interfaces; etc.

22. Other operating system concerns -- secondary storage management; file and directory systems; protection and security; distributed systems; utilities; system administration; etc.

1.5. Telecommunications

23. Telecommunication -- international standards, models, trends; computer networks and control; network design and management; data transmission; line configuration; etc.

24. Telecommunication -- bridges, routers, gateways and system integration, configuration, performance measurement

25. Networks -- architectures and protocols; LANs (topologies, multiplexing, gateways, distributed systems, etc.); WANs (switching and broadcast techniques, routing, etc.); client server; etc.

1.6. Database

26. Database -- DBMS architecture; data models; normalization; integrity; application interface, DDL, DML, query, SQL, etc.; distributed databases; DBMS products; administration; etc.

1.7. Artificial Intelligence

27. Artificial intelligence -- knowledge representation. history, scope, knowledge engineering, inference processing, expert systems, fuzzy logic, CASE-based reasoning, robotics, neural networks, etc.

2.0. Organizational and Management Concepts (Group II)

2.1. General Organization Theory

28. General organization theory -- hierarchical and flow models of organizations, work groups, organizational span, issues pertaining to use of software systems in organizations, etc.

2.2. Information Systems Management

29. Information systems management -- policy; operating procedures; short range and strategic planning; EDP auditing; outsourcing; financial management; customer definition; etc.

30. Other information systems management -- staffing, human resource management, empowerment, staff education and training, staff performance management; etc.

31. Management of information systems sub-functions -- telecommunications, data administration, controlling activities which support software evolution and maintenance; etc.

32. Computer operations management -- tape/DASD management, security, etc.

2.3. Decision Theory

33. Decision theory -- measurement and modeling; decisions under certainty, uncertainty, risk; cost and value of information, decision models, group decision process

2.4. Organizational Behavior

34. Organizational behavior -- job design theory, group dynamics, teamwork, leadership, empowerment, cognitive styles, negotiation and negotiating styles, consensus building, etc.

2.5. and 2.6. (These numbers were not used.)

2.7. Managing the Process of Change

35. Managing the process of change -- reasons for resistance, strategies for motivating change, planning for and managing change

2.8. Legal and Ethical Aspects of IS

36. Legal and ethical aspects -- software sales, licensing, agencies and regulatory bodies, contract fundamentals, privacy law, code of ethics, plagiarism, liability, warranties, misuses, etc.

2.9. Professionalism

37. Professionalism -- historical and social context of computing; certification issues, organizations and conferences, current literature, etc.

2.10. Interpersonal Skills

38. Personal skills -- proactive behavior, goal setting and personal decision making, time management, continuous personal development

39. Interpersonal skills -- oral and written communication, interviewing, listening, presentation, consulting, writing documentation, principle centered leadership, negotiation, etc.

3.0. Theory and Development of Systems (Group III)

3.1. Systems and Information Concepts

40. Systems and information concepts -- general systems theory, components, and relationships; systems control theory and standards; properties of information systems; etc.

3.2. Approaches to Systems Development

41. Approaches to systems development (models and techniques) -- SDLC, iterative enhancement, development with prototyping and packages, process oriented development, etc.

42. Approaches to systems development (systems engineering considerations) -- software process and product life-cycle; system and software generation methods and tools; etc.

3.3. Systems Development Concepts and Methodologies

43. Systems development concepts and methodologies -- organizational and software process modeling; data modeling; types of development; software engineering process and products; etc.

3.4. Systems Development Tools and Techniques

44. Systems development tools and techniques -- CASE; information engineering; Jackson techniques; code, application, and program generators; GDSS; JAD; reusability; tools; etc.

3.5. Application Planning and 3.6. Risk Management

45. Systems development application planning -- infrastructure planning, e.g., hardware, database, and site; planning for operations, architecture, and security; risk management; etc.

3.7. Project Management

46. Project management (organization and management) -- planning; scheduling and milestones; selection of process model; organizational issues; work breakdown structures; staffing; etc.

47. Project management (control)-- planning, cost estimation, risk analysis, resource allocation, reviews, measurement, feedback, communications, ensuring quality, tools

48. Project management (systems and user documentation) -- systems reference manuals, procedures, user documentation, on-line documentation

49. Configuration management (principles, concepts, roles) -- controlling system evolution, ensuring product integrity

50. Configuration management (documentation) -- change and version controls

51. Configuration management (organizational structures, plans, tools)

52. Systems development quality assurance

53. Project tracking and close down -- PERT, Gantt; close down procedures and requirements

3.8. Information and Business Analysis

54. Information and business analysis -- problem identification, relating the application to the enterprise model, requirements determination and specification

3.9. Information Systems Design

55. Information systems design (logical and physical design) -- system design methods and tools; hardware and software tradeoffs; hardware to software and software to software interfaces; performance prediction; modeling techniques and representations; etc.

56. Information systems design (human computer interaction) -- ergonomics; graphical and other user interfaces; architecture of window managers, tool boxes; etc.

57. Information systems design (software requirements) -- principles; types, such as functional and performance; analysis; identification techniques (prototyping, modeling, simulation); communication with customer; tools

58. Information systems design (software specifications) -- objectives; standards; types; formal models; representations; documents (standards, structure, content, users, etc.); techniques: specification of quality attributes; formal specification languages and tools

59. Information systems design (software design) -- principles of design (abstraction, information hiding, reuse, etc.); paradigms for well-understood systems; levels of design; documentation; design of sub-systems; assessment of design quality; languages and tools; methods; etc.

60. Information systems design (software quality assurance) -- issues, definitions, standards, QA as a controlling discipline, factors affecting quality, quality concerns in phases of the SDLC, metrics, plans, documentation, QA project teams, quality and security, etc.

61. Information systems design (software correctness and reliability) --principles, concepts, modeling, methods

62. Information systems design (verification and validation of software quality assurance) -- role and methods, formal models, independent verification and validation teams, tools, reports

63. Information systems design (software implementation) -- relationship of software design to implementation; relationship of implementation language to design principles; tools; other issues (language structures and programming techniques application generators, etc.); etc.

64. **Information systems design (software and hardware system integration) -- methods, plans, tests including incremental testing during development), assessment and documentation of test results, diagnosing system faults, simulation of missing hardware**

65. **Information systems design (software testing) -- role, principles, standards; relationship to QA; methods; levels of testing; plans; audits; limitations; formal models; documentation; tools; test teams; technical reviews; performance and results analysis; reports; etc.**

3.10. Systems Implementation and Testing Strategies

66. **Systems implementation and testing strategies -- systems construction, software systems construction (e.g., programming, unit testing, load module packaging); software integration; systems conversion; training; systems installation; post implementation review; etc.**

3.11. Systems Operation and Maintenance

67. **Systems operation and maintenance -- reverse and reengineering; kinds of maintenance (perceptive, adaptive, corrective); designing software for maintainability; software maintenance techniques (program reading, reverse engineering, etc.); software maintenance models**

3.12. Systems Development for Specific Types of Information Systems

68. **Systems development for specific types of information systems -- transaction processing, management information, group support; decision support; expert; executive support; etc.**

APPENDIX C
STATISTICAL DATA FOR SURVEY RESPONSES
(Count of responses per topic, Means, and Standard Deviation)

TOPIC	SUMMARY KNOWLEDGE TOPICS	CS			IS			SE			ALL		
		COUNT	MEANS	STDDEV	COUNT	MEANS	STDDEV	COUNT	MEANS	STDDEV	COUNT	MEANS	STDDEV
1	Fundamental data presentation and physical representation of digitized information	26	3.54	0.71	105	3.12	1.00	6	3.83	0.41	137	3.23	0.95
2	CPU architectures and computer system components	26	3.46	0.76	105	2.72	1.02	6	2.83	0.75	137	2.86	1.00
3	Multiprocessor architectures	26	2.81	0.90	105	2.21	0.88	6	2.33	0.82	137	2.33	0.90
4	Digital logic and systems	26	2.92	0.89	104	1.44	1.07	6	2.33	0.82	136	1.76	1.18
5	Formal problems and problem solving	26	3.54	0.65	102	3.09	1.05	6	3.50	0.84	134	3.19	0.99
6	Basic data structures	26	3.85	0.46	106	3.22	0.96	6	3.83	0.41	138	3.36	0.91
7	Complex data structures	26	2.77	0.76	105	2.63	0.94	6	3.00	0.89	137	2.67	0.90
8	Abstract data types	26	3.46	0.71	104	2.57	1.13	6	3.50	0.55	136	2.78	1.11
9	File structures and access methods	26	3.69	0.55	106	3.43	0.88	6	3.33	0.52	138	3.47	0.82
10	Sorting and searching data structures and algorithms	25	3.72	0.61	104	2.81	1.06	6	3.33	0.52	135	3.00	1.04
11	Algorithm efficiency, complexity and metrics	26	3.27	1.00	104	1.61	1.17	6	3.00	0.89	136	1.99	1.31
12	Recursive algorithms	26	3.58	0.70	104	1.89	1.33	6	3.50	0.55	136	2.28	1.40
13	Advanced consideration of algorithms	26	2.35	1.02	104	1.35	1.01	6	2.17	0.41	136	1.57	1.07
14	Programming languages	26	3.50	0.76	105	3.54	0.84	6	3.50	0.55	137	3.53	0.81
15	Machine and assembly languages	26	3.04	1.11	105	1.55	1.03	6	3.00	0.89	137	1.89	1.21
16	Design, implementation, and comparison of programming languages	26	3.08	0.89	105	2.23	1.11	6	2.67	0.52	137	2.41	1.10
17	Architecture, goals and structure of operating systems	26	3.23	0.76	104	2.18	1.10	6	2.50	0.55	136	2.39	1.10

TOPIC	SUMMARY KNOWLEDGE TOPICS	CS			IS			SE			ALL		
		COUNT	MEANS	STDDEV	COUNT	MEANS	STDDEV	COUNT	MEANS	STDDEV	COUNT	MEANS	STDDEV
18	Interaction of operating system and hardware architecture	26	3.00	0.89	104	2.08	1.12	6	2.83	0.98	136	2.29	1.13
19	Operating system process management	26	3.27	0.67	104	1.74	1.06	6	2.83	0.75	136	2.07	1.16
20	Memory management	26	3.31	0.62	104	2.13	1.10	6	2.67	0.82	136	2.38	1.12
21	Resource allocation and scheduling	26	2.73	0.78	104	1.78	1.02	6	2.50	0.55	136	1.99	1.04
22	Other operating system concerns	26	3.04	0.60	106	2.44	1.03	6	2.67	1.03	138	2.56	0.98
23	Telecommunication -- international standards...	26	2.69	0.93	106	2.96	1.02	6	2.50	0.55	138	2.89	0.99
24	Telecommunication -- bridges, routers...	26	2.46	0.86	104	2.62	1.08	6	2.67	0.82	136	2.59	1.03
25	Networks	26	2.85	0.73	105	3.09	0.98	6	2.83	0.75	137	3.04	0.93
26	Database	26	3.46	0.65	106	3.59	0.83	6	3.00	0.63	138	3.54	0.80
27	Artificial intelligence	26	2.73	0.92	105	2.29	1.00	6	2.00	0.63	137	2.36	0.99
28	General organization theory	26	2.38	1.10	104	3.23	0.99	6	2.50	0.84	136	3.04	1.06
29	Information systems management	26	2.23	1.24	104	3.22	0.89	6	1.67	1.03	136	2.96	1.08
30	Other information systems management	26	1.73	1.19	103	2.81	1.11	6	1.00	1.10	135	2.53	1.24
31	Management of information systems sub-functions	26	2.19	1.10	105	2.79	1.07	6	1.67	0.82	137	2.63	1.10
32	Computer operations management	26	1.50	1.24	106	2.09	1.16	6	0.67	0.82	138	1.92	1.21
33	Decision theory	26	2.12	1.07	106	2.85	1.02	6	1.33	0.52	138	2.65	1.09
34	Organizational behavior	26	2.15	1.05	106	2.99	0.97	6	1.67	0.52	138	2.78	1.05
35	Managing the process of change	26	1.81	1.10	106	3.01	0.99	6	1.33	1.03	138	2.71	1.15
36	Legal and ethical aspects	26	2.88	0.99	106	2.99	0.97	6	2.50	0.84	138	2.95	0.97
37	Professionalism	26	2.69	1.16	106	2.72	1.17	6	2.17	0.41	138	2.69	1.14
38	Personal skills	26	3.04	1.08	106	3.22	1.11	6	2.33	0.52	138	3.15	1.09

TOPIC	SUMMARY KNOWLEDGE TOPICS	CS			IS			SE			ALL		
		COUNT	MEANS	STDDEV	COUNT	MEANS	STDDEV	COUNT	MEANS	STDDEV	COUNT	MEANS	STDDEV
39	Interpersonal skills	26	3.50	0.81	105	3.69	0.73	6	3.33	0.52	137	3.64	0.74
40	Systems and information concepts	26	2.54	1.21	104	3.26	0.96	6	2.33	0.52	136	3.09	1.04
41	Approaches to systems development (models and techniques)	26	3.23	0.91	106	3.60	0.86	6	3.00	0.89	138	3.51	0.89
42	Approaches to systems development (systems engineering considerations)	26	3.35	0.75	105	3.34	0.82	6	3.33	0.82	137	3.34	0.80
43	Systems development concepts and methodologies	26	3.42	0.81	106	3.43	0.81	6	3.17	0.75	138	3.41	0.80
44	Systems development tools and techniques	26	3.00	0.89	106	3.24	0.93	6	2.50	0.84	138	3.16	0.93
45	Systems development application planning	26	2.50	0.99	106	2.98	0.95	6	2.17	1.17	138	2.86	0.99
46	Project management (organization and management)	26	2.69	1.01	106	3.27	0.90	6	2.83	0.98	138	3.14	0.95
47	Project management (control)	26	2.58	1.06	106	3.07	0.94	6	2.83	1.33	138	2.97	1.00
48	Project management (systems and user documentation)	26	3.04	1.00	106	3.19	0.96	6	3.00	0.89	138	3.15	0.96
49	Configuration management (principles, concepts, roles)	26	2.35	1.16	104	2.35	1.14	6	3.50	0.84	136	2.40	1.15
50	Configuration management (documentation)	26	2.19	1.17	104	2.41	1.14	6	3.33	0.82	136	2.41	1.15
51	Configuration management (organizational structures, plans, tools)	25	2.12	0.88	103	2.35	1.14	6	2.83	0.98	134	2.33	1.10
52	Systems development quality assurance	26	2.38	1.24	104	2.65	1.01	6	2.83	1.17	136	2.61	1.06
53	Project tracking and close down	26	2.50	0.99	105	2.96	1.05	6	2.33	1.03	137	2.85	1.05
54	Information and business analysis	25	2.64	1.25	106	3.55	0.77	6	3.00	1.26	137	3.36	0.96
55	Information systems design (logical and physical design)	26	3.15	1.16	105	3.49	0.79	6	3.67	0.82	137	3.43	0.88
56	Information systems design (human computer interaction)	26	2.81	0.90	104	3.23	0.87	6	3.00	0.89	136	3.14	0.88
57	Information systems design (software requirements)	26	3.31	0.74	105	3.43	0.79	6	3.33	0.82	137	3.40	0.78
58	Information systems design (software specifications)	26	3.31	0.97	105	3.33	0.86	6	3.00	1.26	137	3.31	0.89
59	Information systems design (software design)	26	3.69	0.47	105	3.23	0.92	6	3.67	0.82	137	3.34	0.86

TOPIC	SUMMARY KNOWLEDGE TOPICS	CS			IS			SE			ALL		
		COUNT	MEANS	STDDEV	COUNT	MEANS	STDDEV	COUNT	MEANS	STDDEV	COUNT	MEANS	STDDEV
60	Information systems design (software quality assurance)	26	2.54	1.03	104	2.91	0.91	6	2.83	1.17	136	2.83	0.95
61	Information systems design (software correctness and reliability)	26	3.00	0.85	104	2.86	0.98	6	2.50	1.05	136	2.87	0.96
62	Information systems design (verification and validation of software quality assurance)	26	2.65	1.09	104	2.54	1.03	6	3.00	0.89	136	2.58	1.03
63	Information systems design (software implementation)	26	3.08	0.93	104	2.99	1.00	6	3.00	0.89	136	3.01	0.98
64	Information systems design (software and hardware system integration)	26	2.85	1.01	104	2.92	1.02	6	2.83	0.75	136	2.90	1.01
65	Information systems design (software testing)	26	2.88	0.95	104	3.06	0.95	6	3.50	0.84	136	3.04	0.95
66	Systems implementation and testing strategies	26	2.96	1.08	104	3.11	0.94	6	3.17	0.75	136	3.09	0.96
67	Systems operation and maintenance	26	2.62	1.02	104	2.75	1.08	6	3.17	0.75	136	2.75	1.05
68	Systems development for specific types of information systems	26	2.19	1.17	105	3.15	0.92	6	2.67	1.21	137	2.95	1.04

VITA

VITA

Ronnie Williams was born in Mobile, Alabama, on April 7, 1944. He served in the U.S. Army from 1962 to 1967, in the United States, Vietnam, and Germany. After leaving the Army, he entered the University of South Alabama in 1968, and graduated with a B.S. in geology in 1970. After graduation, he entered U.S. government service and worked in positions with the Department of Defense and the Federal Emergency Management Agency. He retired from the U.S. government in 1989, in Washington, D.C. After returning to Mobile in 1991, he reentered the University of South Alabama. He is currently completing his Master's degree in Computer and Information Sciences.