COMPUTING CURRICULA GUIDELINES
FOR
ASSOCIATE DEGREE PROGRAMS IN

COMPUTING FOR INFORMATION PROCESSING

TWO-YEAR COLLEGE
COMPUTING CURRICULA TASK FORCE

Computing for Information Processing Committee

Suzanne Gill (Chair)	Mount Royal College
Patrice Gapen	Laramie County Community College
Marvin Gore	Mount San Antonio College
Michael Hanna	Front Range Community College
Mary Jo Haught	MJ Services
Thomas Kirk	IBM Corporation
Joyce Currie Little	Towson State University
Meyer Shopkow	Bronx Community College
Rod Southworth	Laramie County Community College

THE ASSOCIATION FOR COMPUTING MACHINERY

1993

ACM Two-Year College Education Committee
and
Task Force Steering Committee

Karl Klee (Chair)	Jamestown Community College
Richard Austing	University of Maryland
Helene Chlopan	University of Kentucky
John Impagliazzo	Hofstra University
Joyce Currie Little	Towson State University

The Association for Computing Machinery
1515 Broadway, 17th Floor

New York, New York 10036
(212) 869-7440

GUIDELINES FOR
COMPUTING FOR INFORMATION PROCESSING

TABLE OF CONTENTS

PART I
CURRICULA STRUCTURE

PART II
CURRICULA KNOWLEDGE UNIT DETAILS

PART III
CURRICULUM IMPLEMENTATION SAMPLES

PART IV
SUPPORTIVE INFORMATION

LIST OF FIGURES

LIST OF TABLES

PART I

CURRICULA STRUCTURE

1.0 INTRODUCTION AND CHARTER

The Association for Computing Machinery (ACM) established preliminary support for the initiation of the development and updating of curricular guidelines for associate degree programs in computing at the two-year colleges. In August, 1990, ACM formed the Two-Year College Computing Curricula Task Force, a group of over thirty educators and industry professionals from across the United States and Canada. The charge for the curriculum work of the Task Force was to examine all computing-related programs appropriate to the associate-degree level of education, over all types of department boundaries and disciplines. The Task Force identified four broad but distinct curricular areas and the area of computing for other disciplines. Five separate committees of the Task Force were formed to investigate the following areas:

Computer Support Services (CSS)
Computing and Engineering Technology (CET)
Computing for Information Processing (CIP)
Computing Sciences (CS)
Computing for Other Disciplines (COD)

Each committee solicited participation by a large number of individuals and produced a separate report. The Executive Report of the Task Force integrates the five reports and identifies commonalities between them. The Executive Report also acknowledges all those who contributed to the work of the Task Force, participated in working sessions, or provided critical reviews.

This report addresses computing for information processing, a curricular area which encompasses the use of computers and computing technology for the information processing functions of public and private organizations. It includes the development, implementation and maintenance of systems to enable the organization to achieve its information processing goals and objectives. It also

covers a broad foundation of knowledge necessary to the understanding of how computing technology will affect the organization's efficiency and impact on its future.

These guidelines should serve the needs of academic institutions offering two-year programs leading to an associate degree, whether offered at a two-year college or within a four-year institution. The format of the guidelines came in part from the ACM/IEEE-CS recent four-year curriculum report, *Computing Curricula 1991* [1].

For several decades, ACM has been a leader in formulating curricula guidelines for the field of computer and information systems, information processing, and computer science. *Curriculum '68* [2] served as the initial definition of the body of knowledge now known as computer science. That area of computing was addressed by updates in 1978 [3] and in 1991.

In the information systems area, ACM was also a leader, with initial reports for graduate level curricula [4], followed by reports for the baccalaureate level [5]. These reports for information systems were updated in 1981 [6, 7]. Guidelines for the associate-level program in computer programming were published in 1981 [8], as were guidelines for a vocational-technical program in data entry operations and computer operations [9]. The momentum for the current ACM support at the two-year level came from an interest in updating and revising the reports of 1981. ACM was also instrumental in the development of curricula for secondary schools, with recommendations for courses and for the certification of teachers [10, 11].

Earlier guidelines developed for the area of computing for information processing were issued by the U.S. Department of Health, Education and Welfare [12, 13]. The Data Processing Management Association (DPMA) also provided guidelines for the baccalaureate level in 1986 [14] and in 1990 [15], the associate degree level in 1986 [16], and the high school level in 1984 [17].

2.0 GOALS AND OVERVIEW OF THIS REPORT

This report is intended to serve as a framework for the development and support of programs in the area of computing for information processing to address the needs for knowledge and skills to work in the information processing component of organizations. It emphasizes professional career education, a foundation for job placement in industry, preparation for transfer for higher degrees, and the

basis for continued learning and advancement. It recognizes that education provides the foundation upon which later training and learning is built. Specific training using specific implementations is vital for continued learning and advancement in the field.

The report will provide a set of curricular and pedagogical considerations that govern the mapping of program requirements into two-year associate degree programs, a collection of subject matter modules called knowledge units, and samples of a curricular mapping of the knowledge units into courses with a sequence of courses leading to the degree.

3.0 CURRICULUM GOALS AND PROFILES OF GRADUATES

Many traditional age students in the two-year degree programs use the courses in the computing information curriculum as a starting point in their career exploration. They discover their capabilities and interests and often form more ambitious career objectives. Often they start at jobs somewhat beneath their objective, and after gaining experience, move into more advanced positions. It is important to offer them a broad outlook at alternatives in job type and level, to allow them flexibility in their career choices.

Students who have already been employed in the field should begin with the course most appropriate to their prior knowledge, as discussed in 8.3. Students in other disciplines may use the two year program to change direction from an existing career outside the information systems component of an organization, as discussed in 9.1. Students already holding an associate degree and working in the field may ask the college to offer special-needs courses related to their employment. Suggestions are given in 9.2.

Students intending to transfer with junior-level status in a baccalaureate degree program need to take a course of study which articulates well with the target institutions. Programs in information systems are currently offered in departments of business or computer science and information systems and in divisions of arts and sciences. The associate degree program should have the flexibility to provide the proper mix of academic and technical courses to allow appropriate exit routes for the purpose of transfer.

3.1 Goals and Objectives

Since the computing for information processing program addresses both the

entry-level needs and future career path needs of its graduates, each institution planning a program in this area should consider at least one and possibly all of the following goals:

- Prepare graduates for local job opportunities in computer applications development related to business information systems.

- Prepare students for eventual transfer to local baccalaureate programs in areas related to information systems, information technology, or computer studies.

- Provide students an opportunity to complete their knowledge requirements for advancement as well as to complete their academic programs at the associate-degree level.

Graduates for tomorrow's career must be knowledgeable about today's application development, have a working knowledge of today's commonly used tools, and recognize that tomorrow's success depends on the ability to adapt to changes. Students must be able to learn independently, because tools will change. The graduate of a computing for information processing program should, therefore, be able to demonstrate use of today's applications development tools, be adaptable enough to learn tomorrow's tools, and be able to understand and use existing systems.

Graduates of the program should be able to carry out the following specific objectives: understand the capabilities and constraints of computers and computing equipment; evaluate and compare software packages and tools; use current software development languages and tools; perform analysis and design for well-defined problems in a business setting; read, write and verify the correctness of computer programs for well-defined problems; communicate in an acceptable oral and written manner; function comfortably in an organizational environment; function comfortably in a work team situation.

The computing for information processing program will develop a graduate's conceptual foundation of information processing and ability to choose and change specializations during professional development. Graduates will learn how to evolve professionally through formal or informal education; effectively communicate using oral, written and presentation techniques in technical and non-technical situations; and be aware of professional responsibilities and the effects of their professional activity on the organization and society.

3.2 Career Opportunities

There is a need for broadly educated individuals who have specific technical skills. They must have the ability to adapt and continue to learn new skills. Students must have the written and verbal communication skills to present ideas; they must work with others during all phases of software development and implementation. Job positions and titles are continually evolving.

Supply and demand data in this area from the Texas Innovation Information Network System [18] show a strong need for personnel over the next five to ten years. Much of the demand is arising from the retirement of those who entered the field when it was new, with perhaps even more of the demand arising from new applications for automation. A recent Canadian study [19] highlights a projected shortage of software workers. The study categorizes two kinds of software workers, namely those in the software industry who are writing commercial software or selling software services, and those who are in-house computing workers. In the case of workers writing commercial software, employees are drawn largely from four-year degree institutions. However, many workers in the other sectors are drawn from the two-year colleges.

Jobs and career paths for graduates of two-year colleges in the computing for information processing area often begin at the operational end of the systems development life cycle, particularly in mainframe environments. Clearly, though, in the business world, there is an accelerating trend toward downsizing enterprise-wide mainframe systems. A client/server environment is rapidly emerging. This is a mid-range environment in which computing capabilities are distributed to end users via extensive networks, and the central-site mainframe becomes a database resource that supports departmental-level and individual workstation computing. As computing power is distributed to users, entry level jobs tend to migrate toward the front end of the systems development life cycle. Entry-level positions may occur in any of several different computer system environments.

Although students will likely be hired for a specific one of the following equipment platforms, they should have some fundamental knowledge about each of them.

> **Platform 1.** The traditional programmer/analyst position in a large main-frame organization developing systems using current programming languages and tools.
>
> Graduates working within this environment would be expected to do

systems development using third-generation languages (3GL), fourth-generation languages (4GL), or computer-assisted software engineering (CASE) technology, as is found in the local employment community. Graduates need to be able to: analyze the processing needs of a functional unit within an organization, construct a systems model within the constraints of the organization's data dictionary/database, and implement the system using the organization's database and systems development tools. They will have a working knowledge of the organization's function and the database and systems development technologies available to assist in those functions. Traditional systems development and maintenance will continue to exist. Specific knowledge will include analysis, design, programming, testing, installation and maintenance of systems using current tools and techniques.

Platform 2. A mid-sized systems environment where traditional systems development work is important, but increased technical knowledge is necessary.

This option will include knowledge of application generators, networks, and applications maintenance activities. Upon graduation, students, in consultation with an end user, will analyze the processing needs of a functional unit within an organization, construct a systems model, and implement the system on the organization's systems environment. Organizations will require a working knowledge of the computer technologies available. Courses should include systems development using tools in a mini-computer or client/server installation similar to those in the local employment community.

Platform 3. An individual personal computer environment for the development of systems for small applications.

Students should have a choice to work in an individual single-station environment or combine this with other opportunities. Courses selected from the curricula for computer support services would enhance and improve this specialization. This option will also include knowledge of maintenance activities for applications, as well as maintenance and support for hardware and software.

Platform 4. A network station environment for the development of applications for shared usage by network users.

Students should be able to do applications development for a network

environment and also work with the equipment itself. Selection of courses from the curricula for computer support services would also enhance and improve this specialization.

Platform 5. An organization where the end-user assumes more of the development and support activity and the graduate will serve in a technical resource or consultant role.

Graduates, in consultation with the end user, should be able to analyze the processing needs of a functional unit within an organization; identify and evaluate alternative solutions - both hardware and software; implement the chosen solution; work with the client to transfer sufficient expertise to continue to run the system. This environment requires a working knowledge of the organization's function, the computer technologies available to assist in those functions, the technical knowledge to evaluate the technology and install or implement the chosen solution, and the communication skills necessary to transfer the knowledge necessary for the normal operation of the solution.

In all cases, there is a large common body of knowledge required. However, appropriate selection of courses must be made for the specific environment. Career advancement toward the front-end of the systems development life cycle or into management positions depends on acquiring experience and/or further education. A bachelor's degree in information systems and/or experience in one or more major business applications areas is usually required for progression into management or senior positions.

3.3 Profiles of Graduates

A student who has completed the computing for information processing career program should, upon graduation, be qualified for employment in a variety of careers. Some of the job categories are: operations, programming, analysis, networking, database systems, and computer graphics. Sample job titles and descriptions may be found in Part IV of this report. Students may begin their specialization in a chosen area, but some of the specializations will require additional studies beyond the two-year degree program.

A student who has completed the transfer program should be able to gain admittance into a baccalaureate degree program in computer information systems with full junior status. The associate-level degree program should be tailored so that the baccalaureate degree program can be completed in two

additional years.

4.0 COMPONENTS OF CURRICULUM DESIGN

The specific design for the curriculum is the responsibility of the educators at each institution. Data about community needs must be obtained for consideration before a program is developed. The program must be flexible enough to incorporate specific tools of interest to local industry, yet may leave detailed knowledge to that industry to be incorporated as training. The prerequisite structure in a given institution must be appropriate and realistic to that community, which may require that a longer time be allowed for completion in areas without sufficient opportunity to enter the program adequately prepared.

4.1 Study of Community Needs

Studies of community demographic information should be done to allow a detailed assessment of need, appropriateness of level, entrance and exit competencies expected, job placement, and opportunities for continued learning at baccalaureate institutions. Advisory boards and alumni boards are critical for development assistance.

4.2 Subject Areas and Knowledge Units

The body of knowledge needed for the local curriculum must be defined. Subject areas are first identified, then units of knowledge belonging to that subject will be documented. These knowledge units (KU) serve as the building blocks for the development of courses which go into a curriculum or a curriculum track. The details of the particular knowledge units defined for the sample curricula are given in Part II of this report.

4.3 Emphasis and Depth

Each knowledge unit is assigned an emphasis category which will show the particular emphasis or style of the instruction. The emphasis is expressed in

terms of the three paradigms of theory, analysis, and design. Theory refers to conceptual knowledge; analysis relates to experimental experience that validates or tests the theory; design refers to the creation of an entity demonstrating the concept. Each knowledge unit is also assigned a depth indicator which will show the depth to which the unit should be covered. The details of emphasis and depth are included in Part III of this report.

5.0 BUILDING THE CURRICULA

The discipline of computing for information processing encompasses the use of computers, computer technology and computing to meet information handling functions in any organization. The foundation of the content consists of principles and techniques, implemented in some specific manufacturer's hardware and software. The expectation of the curriculum is that it will give a student the educational foundation sufficiently rich to allow the graduate to absorb corporate and manufacturer-specific knowledge and skills.

One of the challenges of a two-year program is the limited time interval for introducing material and then building on that knowledge by application to more complex situations. The spiral theory of learning, returning to material in different contexts and at progressively more advanced levels, is well-established in the curriculum through the structure of the course sequence.

5.1 Subject Areas

The body of knowledge required for the computing for information processing discipline consists of nine major subject areas. They range from hard-skills with technical content to soft-skills covering interpersonal communication techniques. The nine subject areas for computing for other disciplines are:

Applications Development Strategies (AD)
This subject area emphasizes the traditional systems development life cycle involving analysis, design, programming (writing procedural code), testing and implementation. This area is primarily mainframe and large application-oriented and may involve the use of commercial software, such as databases. However, the major activity is in-house, custom development using a linear methodology including writing and testing of code.

Applications Generation Strategies (AG)
This subject area emphasizes a prototyping or modelling approach using package software and code generation, or pre-automated processing containing built-in functions as an integral part of the software. Most fourth-generation languages and database management systems application generators fall into this category. It is distinguished from the application development subject area by the absence of coding and code testing activities. The emphasis is on cyclical development and integration of components using automated tools.

Computer Concepts and Fundamentals (CF)
This subject area emphasizes the function and architecture of computer systems hardware and software technologies as applied to information processing. This area stresses the use of the technologies in providing information processing solutions.

Database and Information Retrieval (DB)
This subject area emphasizes database and information retrieval concepts, including logical storage structures, relational models and integrity concerns.

Interpersonal and Communication Techniques (IC)
This subject area emphasizes the following skill areas: presentations, technical writing, technical reading, use of electronic communication technologies, and teamwork. The subject material is incorporated into the computing courses and augments materials typically found in related courses from other areas such as communications and business.

Organizational Behavior (OB)
This subject area emphasizes the following topics: the uses of information technology, related business processes and measures, organizational behavior, regulatory issues, and outside influences. The subject material is incorporated into the computing courses and augments materials typically found in related courses from other areas such as business.

Programming (PR)
This subject area emphasizes the application of programming languages, specification of data types, elements of a block-structured programming language relating to business computing. It includes simple algorithm development. The objective is to develop the student's ability to analyze, design and implement programmed solutions.

Problem Solution Methodologies (PS)
This subject area emphasizes the application of methodologies and strategies to structure solutions relating to business computing. It includes algorithm development, software engineering, and using systems techniques to solve managerial and organizational problems. The objective is to develop the student's ability to analyze, design and implement solutions.

Social, Ethical, and Professional Issues (SP)
This subject area emphasizes the historical and social context of computing, responsibilities of the computing professional, risks and liabilities, and intellectual property.

5.2 Curricula Options

As noted in Section 3.3, entry-level positions usually occur in several different work environments. At present, these cover personal computer, or individual workstation, multi-user system, multi-station computer network, mid-range system, super-mini system, or mainframe system. In the future, the environment may be different. In defining an associate degree program, each institution must evaluate and assess the computer environment in which the graduate must function, and build in the appropriate equipment-specific instruction needed for the expectations and corporate culture of that environment.

With such a wide variety of computing environments and associated courses, it is tempting to design a curriculum to address one area in depth rather than prepare the student with a base knowledge of fundamental principles which apply across all platforms. The curriculum achieves the latter by offering the fundamental knowledge in the required core courses. It achieves the former in the career samples which contain the projects and issues courses appropriate to the students' objectives and the needs of the local community. In addition, it is possible to offer other computing courses as electives if further specialization is desirable.

The computing-related content of the program should be refined and modified by interactions with a local community advisory committee. The use of corporate technical support for instruction can enhance and improve this requirement.

5.3 Requirements and Electives

The sample curricula in Part III of this report have basic knowledge requirements in the use of tools, techniques, and methodologies of applications development, both for custom-made procedural programming solutions and for purchased-package solutions. The required courses provide this foundation.

Elective courses in the career curricular samples allow for the application of knowledge to realistic business problems thereby reinforcing the exit competencies of the curriculum. These electives should be available to the student for specialized study. They may include computing courses found in the other curriculum reports of the ACM Two-Year College Computing Curricula Task Force such as computing and engineering technology, and computer support services.

5.4 Social and Professional Context

Undergraduates need to understand the basic cultural, social, legal and ethical issues inherent in the discipline of computing. They should also understand the history, current situation, and trends in the discipline of computing. They should also understand their individual roles in this process, as well as appreciate the philosophical questions, technical problems, and aesthetic values that have played an important part in the development of the discipline.

Students need to develop the ability to ask serious questions about the social impact of computing and to evaluate alternative answers to those questions. Future practitioners must be able to anticipate the impact of introducing a given product into a given environment. Will that product enhance or degrade the quality of life? What will the impact be upon individuals, groups and institutions? Students need to be aware of the basic legal and intellectual property rights of software and hardware vendors and users. There is also need to appreciate the ethical values which are the basis for those rights. Finally, future practitioners must understand the responsibility that they will bear for what they do, and the consequences of failure to maintain professional standards.

6.0 REQUIREMENTS FROM OTHER DISCIPLINES

Graduates should have a broad general education as well as a strong technical competence. This will be particularly helpful in information processing work.

This section details some of the other skills and competencies which are required.

Graduates must develop good oral and written communication skills. Courses in English grammar, oral communications, and technical writing should be required. Furthermore, writing must be integrated into technical courses and student reports must be neat, grammatically correct and lucid. Students should also be given the opportunity to make oral presentations. It is essential that students comprehend the written material used in the courses and the oral presentations given by instructors and other students. The ability to read and evaluate technical literature is an important skill that will enable students to keep current in their field. Students should be encouraged to read technical publications and to use manufacturer's literature and data books in support of their laboratory assignments.

Students who elect to study computing for information processing should have a thorough background in mathematics which includes two years of algebra. Applied mathematics skills are essential for all computing professionals. These skills include statistical techniques and business mathematics for accounting and finance. In addition, set theory, logic, and reasoning should be emphasized, especially to support the diagnostic problem-solving needs of the analyst-designer. It is highly recommended that students also have a precalculus course to allow them to enter the calculus sequence if they intend to choose a transfer program.

Basic science courses are an important part of a computing curriculum; students should appreciate the application of theory, analysis and design concepts as a fundamental part of any science. Courses should promote the understanding, measurement, and quantitative expression of physical processes. Laboratory work, including experimentation, observation, measurement and report writing, should be a required part of the study of the sciences.

It is important for a computing for information processing student to be aware of general business practices. Communications, both oral and written, should be supplemented by presentation skills. A business approach to problem-solving should be covered in the course work. Specific courses that develop the business skills and knowledge base should be coordinated with the appropriate business department and covered in their courses. Computer-related courses at the exit level of competency should presume knowledge and use of these skills. The business-related content of the program should be refined and modified by interactions with the local community advisory committee. The use of corporate instructor support can enhance and improve this requirement.

Students in the computing for information processing discipline should have additional experiences to help them develop the capacity for critical thinking, problem solving, and professional development. These experiences can be incorporated into the classroom lectures, laboratories, and extracurricular activities. These additional experiences generally fall into the following categories: working independently; working within a team; oral and written communication; familiarization with the profession; site visits; and work experience. Working independently normally evolves out of completing out-of-class assignments in both laboratory and research projects.

Team projects can evolve from laboratory projects, such as those that require group effort and implementation. Such an experience adds breadth, depth, and realism to the curriculum because business and industry require people to perform in this capacity. A team project will also help to develop communication and problem solving skills. Written and oral communication skills are also developed when a student engages in an independent project and must present this project in writing and to their peers in an oral form. Students in this discipline should develop an appreciation for the current literature and be taught to appreciate the difference between scholarly work and the popular press.

Students should become familiar with the computing societies and organizations. They should be encouraged to attend local professional activities and join student chapters of professional organizations. Finally, school-sponsored site visits and work experience for the student are desirable to enhance the classroom activity and apply it to *real-world* situations.

7.0 RESOURCES

This section summarizes the requirements for laboratories, faculty and staff, and institutional support critical for a computing program.

7.1 Program Laboratories

Institutions must provide instruction in computing in a realistic and up-to-date environment. Adequate laboratory facilities are imperative. Laboratory equipment and computers should be similar to that encountered in industry. Furthermore, sufficient equipment must be available to provide each student with the opportunity to become thoroughly familiar with the use and operation of

equipment and computers common to their major field of study. In addition, manuals, equipment catalogs, professional magazines, and journals should supplement the usual library resources. Laboratories need to be well-maintained and supervised so that all components are functional. For example, the presence of a virus in a laboratory prevents students from doing their work.

Laboratories assignments should apply principles to the design, implementation, maintenance, and testing of hardware, software, and communications systems. Laboratory projects should emphasize current computing tools and teach good experimental methods. The laboratory projects augment lecture instruction.

Two distinct types of laboratory sessions, open and closed, are appropriate for the computing for information processing curriculum. Open laboratories allow students to do unscheduled and unsupervised assignments that may involve the use of a computer, software, or hardware. Students can complete such a project on their own time if the facility is available. Conventional programming assignments are most often done in an open laboratory setting. Open laboratories, staffed by trained assistants, must be available for homework and project assignments.

A closed laboratory is one where the instructor or assistant instructor has a dedicated facility for a period of time for instruction. Laboratory projects for these facilities are often projects that require team cooperation, or closely supervised instruction, or specialized hardware or software. Students complete a closed laboratory project by attending a dedicated scheduled session, usually for one to three hours, at a specific facility. Supervision is provided by the instructor or a qualified assistant who is familiar with the details of the assignment. The specialized equipment, software, and/or supervision needed by closed laboratories may prevent them from being completed as open laboratories; some laboratory projects, such as installing equipment or a network, may prevent the facility from being used as an open laboratory.

Instructors should carefully plan laboratory assignment for both open or closed laboratory sessions. Descriptions should include clear statements of purpose and methodology. Laboratory assignments should be realistically designed, so that an average student can complete the work in the allotted time. It is particularly important that adequate facilities are available to support the goals of each laboratory assignment.

Each laboratory assignment should include a written report. This assignment should then be reviewed by the instructor and returned to the student with a grade and/or comments. Some assignments should require presentations to the

class. These assignments should be well-integrated with the subject matter of the lectures.

Access to computing equipment representative of the local business community is required. A networked microcomputer environment with shared peripherals is advisable. An alternative might be a minicomputer environment with terminals and microcomputers sharing system peripherals. A variety of platforms (including mainframe) is highly desirable and might be realistically achieved by arrangements with the local business community.

Access to procedural languages is essential for the coverage of some of the knowledge units. Access to software tools which are representative of what is currently being used by the local business community is also required. Additional specialized equipment and software might be necessary for elective topics.

Provision for updating equipment in response to changing practices and processes is very important. Colleges should have a plan for laboratory improvement. This plan should include the purchase, or lease, of equipment and software, licenses and contracts, and regular maintenance. This plan must also provide adequate technical support for hardware and software rather than expecting faculty to provide this support. The plan must also be reviewed and updated periodically in light of changing student enrollment patterns and new software and hardware.

7.2 Faculty and Staff

The faculty and staff determine the strength of a good program. Full-time faculty are encouraged to have current training in the changing discipline. New faculty members should have at least a master's degree or the equivalent in computer science or information systems. In addition, relevant business experience should be considered an enhancement of qualifications by the college. Since the two-year college serves the community, it should use adjunct faculty working in the profession, as well as faculty in related disciplines to teach some of the courses. Their expertise should provide additional support for the program. The number of full-time faculty must be great enough to provide a breadth of perspective, program continuity, and proper frequency of course offerings. Full-time faculty must also constitute a sufficient nucleus for planning, coordinating, and advising in the program.

Due to rapidly changing technology, faculty members must maintain current

knowledge of their field and understanding of the tasks industry expects their graduates to perform. Faculty members can remain current by active participation in professional societies, continuing education, reading the literature, and periodic returns to graduate school or to industry. The institution should have a planned, adequately funded program for professional development of its faculty as these new curricular recommendations are being implemented. Local, regional, or national workshops may be needed to prepare existing faculty to effectively teach the core computing topics. Grants from government and industry should be sought to help finance these workshops.

7.3 Institutional Support

The institution must provide adequate support for the instructional program including secretarial and technical support, office space, travel funds and released time. Satisfactory procedures and qualified support personnel are required to keep laboratory and instructional equipment in good repair and working order. Faculty office space should provide adequate privacy for student counseling and advisement. Institutional and department budgets should provide travel funding and released time for faculty to attend conferences, workshops, and professional meetings.

In any program where the technology is an integral part of the curriculum, it is critical that there be an ongoing commitment by the college to maintain and modernize instructional equipment on a timely basis. It is vital that students have adequate availability of equipment for scheduled closed and open laboratories during normal operating hours of the day when qualified assistance is available.

Adequate library holdings are important to any program. Current publications, including a variety of journals, are critical to the viability of a computing program. In such a rapidly changing field, a special effort must be made to keep the library holdings current. Faculty should assign work which is dependent upon the library publications and should emphasize the on-going need to remain current in the ever-changing computing field.

8.0 PROGRAM ISSUES

This section summarizes the recommendations on accreditation and the articulation with other institutions and with industry.

8.1 Accreditation

Currently there is no formal accreditation mechanism for programs in the area of information processing at two-year colleges within the United States. Schools of business are often accredited by the *American Assembly of Collegiate Schools of Business* (AACSB). There is no known plan at this time to offer accreditation of programs in information systems. In Canada, one professional society, the Canadian Information Processing Society (CIPS), accredits both two-year and four-year programs in information systems [20].

8.2 Articulation

Articulation issues are critical to the vitality of a two-year college curriculum. For transfer courses, articulation means that written agreements have been made concerning the awarding of credit at particular colleges or universities for courses completed at the two-year college. For career courses and programs, articulation helps to assure that students are adequately prepared to compete for entry-level positions in targeted job titles. In either case, articulation is an activity which must be pursued energetically and maintained vigilantly by two-year colleges with local colleges, universities, businesses and industries.

8.3 Assessment and Evaluation

Recommendations for the design and content of assessment programs are beyond the scope of this report. However, this is an important area for any institution that wishes to maintain quality, up-to-date programs in computing. Assessment criteria can be applied to student outcomes upon graduation or to entering students for appropriate course placement.

Evaluation of a program should be done every few years and is often mandated by the education departments of certain states. Follow-up studies of graduates to determine their success are helpful for evaluation purposes and are often carried out by the institution as a whole.

8.4 Methodology of Instruction

The delivery method needs to evolve from the traditional lecture method to methods which allow the student to become more involved and more active in the learning processes. Examples of other methodologies include: using interactive learning devices, using multimedia in the classroom, researching a subject using networked databases, solving actual campus or business problems in course projects, students learning and presenting new material to the class, team projects for certain closed and open laboratory assignments, exploring the subject through off-campus experiences.

9.0 SERVICE COURSES

This section discusses the role of the computing for information processing faculty to serve the needs of other disciplines within the college, local industry, and community groups.

9.1 Service to Other Disciplines

The faculty are valuable resources to other departments within the college. They are frequently called upon to provide the development and instruction for computer-related courses within the curriculum of another department. Examples include the teaching of specialized courses in graphics for the arts or mass communications disciplines, and the teaching of microcomputer applications, using word processing, database management, spreadsheets, and graphics for students in business-related fields. Sample courses are included in the Computing for Other Disciplines Committee report.

Courses from computing for information processing may also be of service to students in other majors. For example students in the sciences and students in business would benefit from a computer applications course. Being able to offer this service strengthens the position of the department and enhances its role within the structure of the college.

9.2 Needs of Industry and Community

One major mission of the two-year college is to prepare graduates to serve the needs of local industry and the community. In a rapidly changing field, this service is even more important. These courses may be regularly-offered credit-bearing courses. Examples include advanced database or technical courses,

which might serve a local need or be a catalyst for attracting industry to the community. These courses could also be career electives for a computing for information processing program.

A very specific need of a local industry or community group might be best served by a training-oriented, continuing-education course or professional development seminar. An example is the hands-on study of a particular kind of software, such as a project management package. Other courses might include hands-on microcomputer applications training in specific packages.

10.0 CONCLUSION

The recommendations in this report prescribe knowledge units along with a suggested set of courses which show how these knowledge units may be aligned into courses. Individual institutions are encouraged to tailor their own curriculum to their specific program requirements. The use of knowledge units indicates not only topics, but an emphasis and depth of intensity. Course development is not merely the expansion of a subject area. Rather, it is the creative selection from various knowledge units of different subject areas. The manner of selection of knowledge units is a function of each institution, and should result in a modern presentation of courses offered by these colleges.

This report was prepared by the Computing for Information Processing Committee of the Two-Year College Computing Curricula Task Force of the Association for Computing Machinery. The Steering Committee of the Task Force is appreciative of the many individuals and organizations that supported this project. A listing of their names appears in the Executive Report.

PART II

CURRICULA KNOWLEDGE UNIT DETAILS

1.0 OVERVIEW OF KNOWLEDGE UNITS

The fundamental elements upon which courses in a curriculum are formulated are called knowledge units (KU). The format of knowledge units is shown in Figure II-1.

SUBJECT NAME	**APPLICATION DEVELOPMENT STRATEGIES**
KNOWLEDGE UNIT TAG AND NAME	**AD1: SPECIFICATION METHODS**
DESCRIPTION	Overview of the specification methods in common use.
TOPICS	Minimum number of hours: 5 1. Process specification 2. Data specification 3. Object-oriented specification and inheritance 4. Data capture and specification 5. CASE methodologies and tools
PREREQUISITES	CF6, CF7
REQUISITE FOR	AD3, PS2

Figure II-1 Knowledge Unit Format

Each of these knowledge units is grouped by *Subject Name*. Each of the subject areas is divided into a set of knowledge units. A *Knowledge Unit Tag and Name* has the following format:

Letter-1 Letter-2 Integer: Knowledge Unit Name

The two letters identify the subject area name. A number is used for a numerical ordering of the different knowledge units under a given subject name. Code tags are then identified by a knowledge unit name. For example:

AD1: SPECIFICATION METHODS

indicates that the knowledge unit entitled *Specification Methods* is in the subject area *Application Development Strategies*, and it is the first in the set of knowledge units for that subject area. The *Description* section contains a synopsis of the particular knowledge unit. Knowledge units of subject areas are related to other knowledge units within the curriculum, and are divided into *Prerequisite* to and *Requisite For* other knowledge units within the curriculum.

2.0 SUBJECT AREAS AND RELATED KNOWLEDGE UNITS

Each of the subject areas is defined as a collection of knowledge units. This section contains a listing of the knowledge units within each subject area.

Application Development Strategies (AD)

AD1: Specification Methods
AD2: Life Cycle
AD3: Analysis
AD4: Design
AD5: Implementation
AD6: Post Implementation Activities
AD7: Quality Assurance, Security, Standards
AD8: Prototyping
AD9: Human/Machine Interfaces
AD10: Use of System Queues

Application Generation Strategies (AG)

AG1: Common Productivity Software
AG2: Purchased Application Packages
AG3: Fourth Generation Development Methodology
AG4: Application Generation Software
AG5: Front-end CASE Tools
AG6: Back-end CASE Tools

Computer Concepts and Fundamentals (CF)

CF1: Historical Perspective
CF2: Emerging Technologies
CF3: Overview of Hardware
CF4: Overview of Software
CF5: Operating Systems
CF6: Machine Organization
CF7: Process Management
CF8: Memory Management
CF9: Auxiliary Storage
CF10: Machine Representation of Data
CF11: Introduction to Telecommunications
CF12: Applications on Networks
CF13: Security
CF14: System Administration

Database and Information Retrieval (DB)

DB1 File and Physical Database Organization
DB2: Introduction to Databases
DB3: Database Models
DB4: Logical Storage Structures
DB5: Algorithms for Data Manipulations
DB6: Relational Models
DB7: Database Integrity

Interpersonal and Communication Techniques (IC)

IC1: Presentation Skills
IC2: Technical Writing Skills
IC3: Technical Reading Skills
IC4: Electronic Communication Technologies
IC5: Teamwork

Organizational Behavior (OB)

OB1: Uses of Information Technology
OB2: Organization Management Concepts
OB3: Management of Information Services
OB4: Regulatory Issues and Outside Influences

Programming (PR)

PR1: History of Programming Languages
PR2: Elements of a Programming Language
PR3: Sequence Control
PR4: Data Types
PR5: Structured Data Types
PR6: Introduction to Sorting and Searching
PR7: Data Access
PR8: Object-oriented Programming
PR9: Objects
PR10: Language Translation Systems
PR11: Programming Paradigms

Problem Solution Methodologies (PS)

PS1: Quantitative Methods
PS2: Algorithm Development
PS3: Evaluation of Effectiveness
PS4: Overview of Requirements Specification
PS5: Verification and Validation
PS6: Documentation
PS7: Introduction to Qualitative Methods
PS8: Decision Analysis
PS9: Problem Analysis in Computer Systems
PS10: Potential Problem Analysis
PS11: Project Management

Social, Ethical, and Professional Issues (SP)

SP1: Evolution of Computing and the Professional
SP2: Intellectual Property
SP3: Software Protection and Security
SP4: System Security
SP5: Social Responsibility of Professionals
SP6: Data Collection and Privacy
SP7: Risks in Large Systems

3.0 DETAILS OF KNOWLEDGE UNITS

This section contains the details of the knowledge units within the subject areas. The knowledge units are listed two per page by subject area in the format discussed in Section 1.1

SUBJECT NAME **APPLICATION DEVELOPMENT STRATEGIES**

KNOWLEDGE UNIT TAG AND NAME **AD1: SPECIFICATION METHODS**

DESCRIPTION Overview of the specification methods in common use.

TOPICS Minimum number of hours: 5
1. Process specification
2. Data specification
3. Object-oriented specification and inheritance
4. Data capture and specification
5. Computer-Assisted Software Engineering (CASE) methodologies and tools

PREREQUISITE CF6, CF7

REQUISITE FOR AD3

SUBJECT NAME **APPLICATION DEVELOPMENT STRATEGIES**

KNOWLEDGE UNIT TAG AND NAME **AD2: LIFE CYCLE**

DESCRIPTION Overview of the concept of the system life cycle.

TOPICS Minimum number of hours: 2
1. Steps in the life cycle
2. Integration and interaction of steps
3. Deliverables and documentation
4. Impact of the development activity on the organization

PREREQUISITES AD1

REQUISITE FOR AD3, AD8

SUBJECT NAME **APPLICATION DEVELOPMENT STRATEGIES**

KNOWLEDGE UNIT TAG AND NAME **AD3: ANALYSIS**

DESCRIPTION The ability to analyze requirements using a variety of information gathering techniques.

TOPICS Minimum number of hours: 10

1. Information gathering techniques
2. Information organizing techniques
3. Prepare a feasibility study
4. Prepare a project proposal or functional specification.
5. Assessment of the consequences of change

PREREQUISITE IC2, IC3, IC5, OB2

REQUISITE FOR AG3

SUBJECT NAME **APPLICATION DEVELOPMENT STRATEGIES**

KNOWLEDGE UNIT TAG AND NAME **AD4: DESIGN**

DESCRIPTION Knowledge of formal approaches to design.

TOPICS Minimum number of hours: 14

1. Design of data structures
2. Operations on data structures
3. Specifications for testing and debugging using rigorous formal approaches
4. Designing for maintenance - robustness, reliability, security, and recovery
5. Designing for performance
6. Assessment of the consequences of change

PREREQUISITE PS2

REQUISITE FOR AD5, AD9

SUBJECT NAME **APPLICATION DEVELOPMENT STRATEGIES**

KNOWLEDGE UNIT TAG AND NAME **AD5: IMPLEMENTATION**

DESCRIPTION The process of transforming the specification into the deliverable.

TOPICS Minimum number of hours: 13

1. Choice of construction techniques, including languages and programming paradigms such as object-oriented and data-driven
2. Data conversion
3. Procedure conversion
4. Program verification and correctness
5. Training
6. Interface and performance testing
7. Component re-use and portability

PREREQUISITE AD4, PS3, PS5

REQUISITE FOR AD6

SUBJECT NAME **APPLICATION DEVELOPMENT STRATEGIES**

KNOWLEDGE UNIT TAG AND NAME **AD6: POST IMPLEMENTATION ACTIVITIES**

DESCRIPTION Knowledge of the production part of the systems activity, including on-going maintenance and enhancements.

TOPICS Minimum number of hours: 4

1. Different kinds of maintenance
2. Audits and reviews
3. Management of change and its impact
4. Awareness of CASE tools to aid in reengineering or restructuring of existing applications

PREREQUISITE AD5

REQUISITE FOR None

SUBJECT NAME **APPLICATION DEVELOPMENT STRATEGIES**

KNOWLEDGE UNIT TAG AND NAME **AD7: QUALITY ASSURANCE, SECURITY AND STANDARDS**

DESCRIPTION Incorporation of quality assurance, security, and standards throughout the life cycle.

TOPICS Minimum number of hours: 6

1. Testing, debugging, generation of test data and change procedures
2. Security topics, including encryption, decryption, passwords (user,system, file and check points), and data backup
3. Standards, including ANSI, ISO, file naming conventions, directory naming conventions, documentation types and standards
4. Walkthroughs and other reviews

PREREQUISITES AD1

REQUISITE FOR AD8, AD4

SUBJECT NAME **APPLICATION DEVELOPMENT STRATEGIES**

KNOWLEDGE UNIT TAG AND NAME **AD8: PROTOTYPING**

DESCRIPTION A discussion of the techniques available for the rapid development of critical systems components in an effort to test the feasibility of a portion, often leaving out details.

TOPICS Minimum number of hours: 8

1. Different prototyping levels
2. Different prototyping deliverables, such as requirements, design, functionality, and performance

PREREQUISITES AD1, AD7, AD9

REQUISITE FOR AG3

SUBJECT NAME **APPLICATION DEVELOPMENT STRATEGIES**

KNOWLEDGE UNIT TAG AND NAME **AD9: HUMAN/MACHINE INTERFACES**

DESCRIPTION This should include the two areas of physical interfaces (ergonomics and the cognitive).

TOPICS Minimum number of hours: 3
1. Menu driven systems
2. Graphical user interfaces (GUI)
3. Response time requirements

PREREQUISITES AD3

REQUISITE FOR AD8, AG3

SUBJECT NAME **APPLICATION DEVELOPMENT STRATEGIES**

KNOWLEDGE UNIT TAG AND NAME **AD10: USE OF SYSTEM QUEUES**

DESCRIPTION Subsystems are used to more effectively divide the functionality of differing applications into discrete units and to better enhance security. Students need to understand that different subsystems provide different environments and understand how to use the various queues to manage the processing.

TOPICS Minimum number of hours: 4
1. Subsystems (commonly available, how security may be enhanced, effective allocation of system resources)
2. Interactive job queues
3. Batch job queues
4. Message queues and print queues

PREREQUISITES CF7

REQUISITE FOR AD9

SUBJECT NAME **APPLICATION GENERATION STRATEGIES**

KNOWLEDGE UNIT TAG AND NAME **AG1: COMMON PRODUCTIVITY SOFTWARE**

DESCRIPTION Common productivity software which enhance productivity through the use of generalized packages not specific to information processing. Use of such software in information processing.

TOPICS Minimum number of hours: 10

1. Word Processing
2. Spreadsheet
3. Graphics
4. Database
5. Other tools such as CASE and Rapid Application Development (RAD)

PREREQUISITES CF1

REQUISITE FOR AD3, AG2

SUBJECT NAME **APPLICATION GENERATION STRATEGIES**

KNOWLEDGE UNIT TAG AND NAME **AG2: PURCHASED APPLICATION PACKAGES**

DESCRIPTION Selection and installation of commercial available software.

TOPICS Minimum number of hours: 6

1. Advantages/disadvantages of packages
2. Integration and tradeoffs
3. Selection process

PREREQUISITE AD2, AD9, PS1, PS7

REQUISITE FOR None

SUBJECT NAME **APPLICATION GENERATION STRATEGIES**

KNOWLEDGE UNIT TAG AND NAME **AG3: FOURTH GENERATION DEVELOPMENT METHODOLOGY**

DESCRIPTION An overview of the various methodologies used in the generation of applications using fourth-generation languages rather than the traditional systems development life cycle and third-generation languages.

TOPICS Minimum number of hours: 5

1. Use of prototyping
2. End-user involvement
3. Appropriate development language and/or application package selection

PREREQUISITES AG1, DB2

REQUISITE FOR AG4

SUBJECT NAME **APPLICATION GENERATION STRATEGIES**

KNOWLEDGE UNIT TAG AND NAME **AG4: APPLICATION GENERATION SOFTWARE**

DESCRIPTION Use of software (commonly called application generators) to develop applications efficiently and effectively.

TOPICS Minimum number of hours: 9

1. Use of a DBMS, including a data dictionary
2. Report and screen generators
3. Query languages such as SQL
4. Procedural languages such as C++
5. Non-procedural languages (4GL's)
6. Processing considerations

PREREQUISITES AG3, DB5, DB6

REQUISITE FOR AG5

SUBJECT NAME **APPLICATION GENERATION STRATEGIES**

KNOWLEDGE UNIT TAG AND NAME **AG5: FRONT-END CASE TOOLS**

DESCRIPTION Awareness of the use of CASE tools to assist in the systems analysis and design phases of application development.

TOPICS Minimum number of hours: 1
1. Data Flow Diagrams
2. Prototyping
3. Project management

PREREQUISITES AG4

REQUISITE FOR AG6

SUBJECT NAME **APPLICATION GENERATION STRATEGIES**

KNOWLEDGE UNIT TAG AND NAME **AG6: BACK-END CASE TOOLS**

DESCRIPTION Awareness of the use of CASE tools to assist in the implementation phase of application development.

TOPICS Minimum number of hours: 1
1. Code generation (based on analyst-generated specifications)
2. Quality assurance

PREREQUISITES AG5

REQUISITE FOR None

SUBJECT NAME **COMPUTER CONCEPTS AND FUNDAMENTALS**

KNOWLEDGE UNIT TAG AND NAME **CF1: HISTORICAL PERSPECTIVE**

DESCRIPTION An introduction to the historical context in which the discipline of computing exists, with an emphasis on how quickly the discipline has evolved and currently interacts with and serves the various interests of our society.

TOPICS Minimum number of hours: 1

1. The early roots of computing
2. Impact on society
3. Career choices
4. Impact of change and life-long learning

PREREQUISITES Computer literacy or equivalent

REQUISITE FOR CF2

SUBJECT NAME **COMPUTER CONCEPTS AND FUNDAMENTALS**

KNOWLEDGE UNIT TAG AND NAME **CF2: EMERGING TECHNOLOGIES**

DESCRIPTION Current technology serves as a starting point for the study of future technologies within the computing industry.

TOPICS Minimum number of hours: 2

Awareness of:

1. Distributed processing
2. Multimedia
3. Artificial intelligence and expert systems
4. New system development methodologies such as object oriented
5. New tools and techniques
6. New architectures such as RISC and parallel

PREREQUISITES CF1

REQUISITE FOR None

SUBJECT NAME **COMPUTER CONCEPTS AND FUNDAMENTALS**

KNOWLEDGE UNIT TAG AND NAME **CF3: OVERVIEW OF HARDWARE**

DESCRIPTION An overview of computer hardware used in information processing.

TOPICS Minimum number of hours: 1
1. The characteristics of principal system components
2. Input/output devices and interfaces
3. Types of storage

PREREQUISITES CF1

REQUISITE FOR AD1

SUBJECT NAME **COMPUTER CONCEPTS AND FUNDAMENTALS**

KNOWLEDGE UNIT TAG AND NAME **CF4: OVERVIEW OF SOFTWARE**

DESCRIPTION An overview of software products used in information processing.

TOPICS Minimum number of hours: 2
1. Operating systems
2. Utilities
3. Applications software
4. Distinction between systems and applications software

PREREQUISITES CF1

REQUISITE FOR AD1, CF5

SUBJECT NAME **COMPUTING CONCEPTS AND FUNDAMENTALS**

KNOWLEDGE UNIT TAG AND NAME **CF5: OPERATING SYSTEMS**

DESCRIPTION Overview of operating systems.

TOPICS Minimum number of hours: 2
Characteristics of, and distinctions between operating systems for:

1. Microcomputer
2. Minicomputer
3. Mainframe
4. Network operating systems

PREREQUISITES CF4

REQUISITE FOR CF6, AD1

SUBJECT NAME **COMPUTING CONCEPTS AND FUNDAMENTALS**

KNOWLEDGE UNIT TAG AND NAME **CF6: MACHINE ORGANIZATION**

DESCRIPTION Overview of major systems components and their interaction.

TOPICS Minimum number of hours: 1

1. Basic system organization and interaction
2. Instruction representation
3. Control unit, instruction fetch and execution, operand fetch

PREREQUISITES CF3, CF5

REQUISITE FOR CF7, PR2

SUBJECT NAME **COMPUTING CONCEPTS AND FUNDAMENTALS**

KNOWLEDGE UNIT TAG AND NAME **CF7: PROCESS MANAGEMENT**

DESCRIPTION Awareness of the control of the major activities within the CPU. Including work management queues commonly used to hold jobs, messages and printer output until system resources are available or until users wish to further process the information.

TOPICS Minimum number of hours: 1

1. Processes
2. Scheduling
3. Multitasking/Multiprocessing
4. Interactive job queues
5. Batch job, message, and print queues

PREREQUISITES CF6

REQUISITE FOR PR10, AD10, CF8

SUBJECT NAME **COMPUTING CONCEPTS AND FUNDAMENTALS**

KNOWLEDGE UNIT TAG AND NAME **CF8: MEMORY MANAGEMENT**

DESCRIPTION Overview of the diversity of memory organization and management schemes.

TOPICS Minimum number of hours: 1

1. Overview of memory addressing
2. Overview of primary, virtual, extended, expanded and cache memory

PREREQUISITES CF7

REQUISITE FOR PR5

SUBJECT NAME **COMPUTING CONCEPTS AND FUNDAMENTALS**

KNOWLEDGE UNIT TAG AND NAME **CF9: AUXILIARY STORAGE**

DESCRIPTION Overview of the diversity of auxiliary storage media, their operational characteristics, and their storage capacities and retrieval methods. Examples are: tape, floppy disk, hard disk, laser optical disks.

TOPICS Minimum number of hours: 3
1. Devices - disk, tape, CD-ROM
2. Files and directories

PREREQUISITES CF3, CF4

REQUISITE FOR DB1

SUBJECT NAME **COMPUTING CONCEPTS AND FUNDAMENTALS**

KNOWLEDGE UNIT TAG AND NAME **CF10: MACHINE REPRESENTATION OF DATA**

DESCRIPTION Methods of representing numeric and non-numeric data in a machine.

TOPICS Minimum number of hours: 3
1. Binary and hexadecimal number systems
2. Unsigned, signed, floating point, integers
3. Alphanumeric data
4. Impact on instruction execution

PREREQUISITES CF6

REQUISITE FOR PR4

SUBJECT NAME **COMPUTER CONCEPTS AND FUNDAMENTALS**

KNOWLEDGE UNIT TAG AND NAME **CF11: INTRODUCTION TO TELECOMMUNICATIONS**

DESCRIPTION An overview of telecommunications hardware and software.

TOPICS Minimum number of hours: 3
1. Elements of data communications
2. Network architecture
3. Network topologies
4. Standards and protocols

REREQUISITES CF3, CF4

REQUISITE FOR AD1

SUBJECT NAME **COMPUTING CONCEPTS AND FUNDAMENTALS**

KNOWLEDGE UNIT TAG AND NAME **CF12: APPLICATIONS ON NETWORKS**

DESCRIPTION A discussion of the applications and functions generally supported on networks or network nodes.

TOPICS Minimum number of hours: 3
1. Electronic/voice/multi-media mail
2. File sharing
3. Dial-in access
4. Resource sharing
5. Software control
6. Backups, security and authentication

7. Awareness of administration

PREREQUISITES CF11

REQUISITE FOR DB7, AD3

SUBJECT NAME **COMPUTING CONCEPTS AND FUNDAMENTALS**

KNOWLEDGE UNIT TAG AND NAME **CF13: SECURITY**

DESCRIPTION Awareness of the importance of achieving secure computing and communication systems, the difficulties of achieving security in increasingly open systems environments. Worms, viruses, vaccines, penetrating operating systems, security procedures, encryption techniques.

TOPICS Minimum number of hours: 1

1. Authentication (access, security, passwords)
2. Encryption
3. Viruses
4. Operating system penetration

PREREQUISITES CF12

REQUISITE FOR DB7, AD3

SUBJECT NAME **COMPUTER CONCEPTS AND FUNDAMENTALS**

KNOWLEDGE UNIT TAG AND NAME **CF14: SYSTEM ADMINISTRATION**

DESCRIPTION An overview of systems administration functions.

TOPICS Minimum number of hours: 3

1. Backup and recovery
2. Batch vs. interactive processing
3. Exception handling
4. End user computing support
5. Data administration
6. Security and disaster planning

PREREQUISITES CF12

REQUISITE FOR AD3

SUBJECT NAME **DATABASE AND INFORMATION RETRIEVAL**

KNOWLEDGE UNIT TAG AND NAME **DB1: FILE AND PHYSICAL DATABASE ORGANIZATION**

DESCRIPTION Fundamental file organizations and access methods.

TOPICS Minimum number of hours: 4
1. Sequential file organization and access methods
2. Indexed file organization and access methods
3. Random file organization and access methods
4. Blocking and buffering

PREREQUISITES CF9

REQUISITE FOR DB4

SUBJECT NAME **DATABASE AND INFORMATION RETRIEVAL**

KNOWLEDGE UNIT TAG AND NAME **DB2: INTRODUCTION TO DATABASES**

DESCRIPTION An introduction to database systems, including a survey of emerging technologies and their impact.

TOPICS Minimum number of hours: 1
1. History
2. Database components (data dictionary, DBMS, users, administrator)
3. Applications and impact on society

PREREQUISITES CF4

REQUISITE FOR DB3

SUBJECT NAME **DATABASE AND INFORMATION RETRIEVAL**

KNOWLEDGE UNIT TAG AND NAME **DB3: DATABASE MODELS**

DESCRIPTION An introduction to database models, and their application.

TOPICS Minimum number of hours: 4
1. History (hierarchical, network)
2. Conceptual database models (relational, object-oriented)
3. Data normalization
4. Applications

PREREQUISITES DB2

REQUISITE FOR DB4

SUBJECT NAME **DATABASE AND INFORMATION RETRIEVAL**

KNOWLEDGE UNIT TAG AND NAME **DB4: LOGICAL STORAGE STRUCTURES**

DESCRIPTION Identification of file organizations and data structures that support database systems.

TOPICS Minimum number of hours: 2
1. File organization (sequential, random, indexed)
2. Data structures (trees, graphs, B-trees, inverted lists)
3. Hashing functions

PREREQUISITES DB3, CF9

REQUISITE FOR DB5, DB6

SUBJECT NAME **DATABASE AND INFORMATION RETRIEVAL**

KNOWLEDGE UNIT TAG AND NAME **DB5: ALGORITHMS FOR DATA MANIPULATIONS**

DESCRIPTION Identification of fundamental algorithms used in managing a database.

TOPICS Minimum number of hours: 4
1. Record and file creation, update, deletion
2. Searching, joining records
3. Merging ordered files
4. Sorts

PREREQUISITES DB4, PR5

REQUISITE FOR DB7

SUBJECT NAME **DATABASE AND INFORMATION RETRIEVAL**

KNOWLEDGE UNIT TAG AND NAME **DB6: RELATIONAL MODELS**

DESCRIPTION An overview of the relational database model and query languages.

TOPICS Minimum number of hours: 5
1. Terminology
2. Relational calculus
3. Normal forms and tradeoffs
4. Query languages

PREREQUISITES DB4, PR3

REQUISITE FOR DB7

SUBJECT NAME **DATABASE AND INFORMATION RETRIEVAL**

KNOWLEDGE UNIT TAG AND NAME **DB7: DATABASE INTEGRITY**

DESCRIPTION Identification of controls that help assure integrity of databases.

TOPICS Minimum number of hours: 2
1. Security, access control
2. Privacy, encryption
3. Concurrency, deadlock
4. Backups and fault recovery

PREREQUISITES DB5

REQUISITE FOR None

SUBJECT NAME **INTERPERSONAL AND COMMUNICATION TECHNIQUES**

KNOWLEDGE UNIT TAG AND NAME **IC1: PRESENTATION SKILLS**

DESCRIPTION Use of presentation skills required to analyze, develop, and implement computer information systems.

TOPICS Minimum number of hours: 1
1. Application of public speaking skills
2. Appropriate use of presentation technologies (including visual media)
3. Interviewing skills (including evaluation and feedback)
4. End-user training techniques (including change management)

5. Includes interaction in public forums to support and promote IS activities

PREREQUISITES AD3

REQUISITE FOR IC2, IC5

SUBJECT NAME **INTERPERSONAL AND COMMUNICATION TECHNIQUES**

KNOWLEDGE UNIT TAG AND NAME **IC2: TECHNICAL WRITING SKILLS**

DESCRIPTION Application of technical writing skills required to analyze, develop and implement computer information systems.

TOPICS Minimum number of hours: 4

1. Creating analysis and design documents, including feasibility studies, cost/benefit analyses, RFP's (Request for Proposal), RFQ's (Request for Quotation)
2. System specification documents
3. Creating final systems, programming and operational documentation, and end-user documentation

PREREQUISITES IC1

REQUISITE FOR IC4, OB3, AD3

SUBJECT NAME **INTERPERSONAL AND COMMUNICATION TECHNIQUES**

KNOWLEDGE UNIT TAG AND NAME **IC3: TECHNICAL READING SKILLS**

DESCRIPTION Ability to locate, evaluate and assimilate professional publications.

TOPICS Minimum number of hours: 2

1. Research, obtain, and utilize current IS literature
2. Assimilate and use technical materials including hardware and software specifications and manuals

PREREQUISITES None

REQUISITE FOR OB3, AD3, PS6

SUBJECT NAME **INTERPERSONAL AND COMMUNICATION TECHNIQUES**

KNOWLEDGE UNIT TAG AND NAME **IC4: ELECTRONIC COMMUNICATION TECHNOLOGIES**

DESCRIPTION Awareness of electronic technologies to facilitate communications in the professional environment.

TOPICS Minimum number of hours: 2
Use and integration of:
1. Electronic mail
2. Voice mail
3. Facsimile transmission

PREREQUISITES IC2, IC5

REQUISITE FOR None

SUBJECT NAME **INTERPERSONAL AND COMMUNICATION TECHNIQUES**

KNOWLEDGE UNIT TAG AND NAME **IC5: TEAMWORK**

DESCRIPTION Ability to work effectively in a team or work group.

TOPICS Minimum number of hours: 6
1. Ability to recognize and interact appropriately using various individual cognitive styles
2. Understanding of group dynamics, including group behavior
3. The decision-making process and power bases

4. Integration of leadership skills, creativity, and individual participation in productive information systems group activities

PREREQUISITES IC1

REQUISITE FOR IC4, OB3, AD3

SUBJECT NAME **ORGANIZATIONAL BEHAVIOR**

KNOWLEDGE UNIT TAG AND NAME **OB1: USES OF INFORMATION TECHNOLOGY**

DESCRIPTION An overview of types of systems in a organization.

TOPICS Minimum number of hours: 3
1. Transaction processing
2. Decision support systems
3. Management information systems
4. Expert systems

PREREQUISITES CF1

REQUISITE FOR OB2, OB4

SUBJECT NAME **ORGANIZATIONAL BEHAVIOR**

KNOWLEDGE UNIT TAG AND NAME **OB2: ORGANIZATION MANAGEMENT CONCEPTS**

DESCRIPTION An overview of managerial problems in technological innovation.

TOPICS Minimum number of hours: 1
1. Risk management
2. Planning and control of resources, including staff selection and development, motivation, and appraisal
3. Applied management techniques

PREREQUISITES OB1

REQUISITE FOR OB3

SUBJECT NAME **ORGANIZATIONAL BEHAVIOR**

KNOWLEDGE UNIT TAG AND NAME **OB3: MANAGEMENT OF INFORMATION SERVICES**

DESCRIPTION An overview of management of information services in support of the business organization.

TOPICS Minimum number of hours: 1
Awareness of the impact of:
1. Installation and facilities management
2. Contract services
3. Information Systems Department organization and control
4. Resource budgeting and allocation
5. Selection and implementation of methods and standards for application development

PREREQUISITES OB2, IC2, IC3, IC5

REQUISITE FOR None

SUBJECT NAME **ORGANIZATIONAL BEHAVIOR**

KNOWLEDGE UNIT TAG AND NAME **OB4: REGULATORY ISSUES AND OUTSIDE INFLUENCES**

DESCRIPTION An overview of the effect of regulation and outside controls.

TOPICS Minimum number of hours: 1
1. Effects of budgetary constraints
2. Effects of legal issues and changes in the legal environment
3. Effects of mergers and acquisitions
4. Corporate planning changes

5. Competition and shifts in the economy

PREREQUISITES OB1

REQUISITE FOR None

SUBJECT NAME **PROGRAMMING**

KNOWLEDGE UNIT TAG AND NAME **PR1: HISTORY OF PROGRAMMING LANGUAGES**

DESCRIPTION The history of early programming languages and the evolution of procedural languages.

TOPICS Minimum number of hours: 1
1. Early languages (FORTRAN, ALGOL)
2. Development of procedural languages such as Pascal

PREREQUISITES None

REQUISITE FOR PR2

SUBJECT NAME **PROGRAMMING**

KNOWLEDGE UNIT TAG AND NAME **PR2: ELEMENTS OF A PROGRAMMING LANGUAGE**

DESCRIPTION An introduction to the syntax, logic and concepts of a contemporary block-structured programming language. (Prior knowledge of programming is desireable.)

TOPICS Minimum number of hours: 7
1. Type declarations, names, and variables
2. Arithmetic operators and assignment statements
3. Selection and iteration
4. Subprograms with parameters
5. Functions with parameters
6. Arrays

PREREQUISITES PR1

REQUISITE FOR PR3, PR8, PR10, PR11, PS2

SUBJECT NAME **PROGRAMMING**

KNOWLEDGE UNIT TAG AND NAME **PR3: SEQUENCE CONTROL**

DESCRIPTION Sequence of operations in evaluating expressions and implementing statements, including exception handling.

TOPICS Minimum number of hours: 4
1. Expression evaluation
2. Simple and compound statements
3. User-defined subprograms as expression and statement extensions
4. Defensive programming, error recovery and interrupt handling.

PREREQUISITES PR2

REQUISITE FOR DB6, PR4, PR6, PR7

SUBJECT NAME **PROGRAMMING**

KNOWLEDGE UNIT TAG AND NAME **PR4: DATA TYPES**

DESCRIPTION Atomic versus structured data types. An introduction to user-defined data types and applications.

TOPICS Minimum number of hours: 4
1. Choice and representation of data types
2. User-defined data types with applications

PREREQUISITES PR2

REQUISITE FOR PR5

SUBJECT NAME **PROGRAMMING**

KNOWLEDGE UNIT TAG AND NAME **PR5: STRUCTURED DATA TYPES**

DESCRIPTION An introduction to structured data types provided in most procedural languages. Emphasis on specification, set of values, set of operations, and representation. Understanding of fundamental data structures; concepts of abstract data types.

TOPICS Minimum number of hours: 4
1. Arrays, strings, and records
2. Stacks, queues, linked lists, and trees
3. Definition of abstract data types
4. Information hiding

PREREQUISITES PR4

REQUISITE FOR DB4, AD4

SUBJECT NAME **PROGRAMMING**

KNOWLEDGE UNIT TAG AND NAME **PR6: INTRODUCTION TO SORTING AND SEARCHING**

DESCRIPTION An introduction to sorting and searching algorithms.

TOPICS Minimum number of hours: 1
1. Linear search of an array
2. Sorting an array (for example, selection sort)

PREREQUISITES PR3

REQUISITE FOR DB5

SUBJECT NAME **PROGRAMMING**

KNOWLEDGE UNIT TAG AND NAME **PR7: DATA ACCESS**

DESCRIPTION The mechanisms for providing access to data, and control of those mechanisms.

TOPICS Minimum number of hours: 4
1. Static and dynamic scope
2. Parameter passing mechanisms
3. Type checking, including static, dynamic, untyped, explicit, and implicit
4. Run-time storage management

PREREQUISITES PR3

REQUISITE FOR AD4

SUBJECT NAME **PROGRAMMING**

KNOWLEDGE UNIT TAG AND NAME **PR8: OBJECT-ORIENTED PROGRAMMING**

DESCRIPTION An introduction to the objected-oriented paradigm.

TOPICS Minimum number of hours: 1
1. Encapsulation
2. Methods
3. Messaging
4. Languages

PREREQUISITES CF2, PR2

REQUISITE FOR PR9

SUBJECT NAME **PROGRAMMING**

KNOWLEDGE UNIT TAG AND NAME **PR9: OBJECTS**

DESCRIPTION Object-oriented data types.

TOPICS Minimum number of hours: 1
1. Examples of objects in an object-oriented language
2. Advantages and disadvantages of using objects

PREREQUISITES PR8

REQUISITE FOR None

SUBJECT NAME **PROGRAMMING**

KNOWLEDGE UNIT TAG AND NAME **PR10: LANGUAGE TRANSLATION SYSTEMS**

DESCRIPTION Overview of programming language translation processes.

TOPICS Minimum number of hours: 2
1. Lexical analysis, symbol tables
2. Interpreters
3. Assemblers and linkers
4. Compilers, code generation and optimization

PREREQUISITES CF4, CF5, PR2

REQUISITE FOR None

SUBJECT NAME **PROGRAMMING**

KNOWLEDGE UNIT TAG AND NAME **PR11: PROGRAMMING PARADIGMS**

DESCRIPTION Non-procedural paradigms and languages, including applications of each and comparison with procedural programming.

TOPICS Minimum number of hours: 1
1. Functional languages
2. Logic languages

PREREQUISITES PR2

REQUISITE FOR None

SUBJECT NAME **PROBLEM SOLUTION METHODOLOGIES**

KNOWLEDGE UNIT TAG AND NAME **PS1: QUANTITATIVE METHODS**

DESCRIPTION Application of quantitative methodologies used to develop systems. The objective of applying mathematical techniques is to demonstrate ways *real* problems can be modeled and reduced to solvable components.

TOPICS Minimum number of hours: 1
Awareness of the use of:
1. Statistical analysis
2. Cost benefit analysis
3. Linear programming

PREREQUISITES PS2

REQUISITE FOR AD1

SUBJECT NAME **PROBLEM SOLUTION METHODOLOGIES**

KNOWLEDGE UNIT TAG AND NAME **PS2: ALGORITHM DEVELOPMENT**

DESCRIPTION A detailed knowledge of a systematic approach to developing effective solutions; the ability to apply at least one systematic algorithm design method (e.g., data driven, object oriented, structured methods) and use of automated tools where possible.

TOPICS Minimum number of hours: 6

1. Introduction to algorithmic problem solving
2. Control structures: sequence, selection, iteration
3. Standard data types and their uses
4. Procedural abstraction, parameters
5. Order of mathematical processing, and heuristic methods

PREREQUISITES Computer literacy or equivalent

REQUISITE FOR PR2, AD1

SUBJECT NAME **PROBLEM SOLUTION METHODOLOGIES**

KNOWLEDGE UNIT TAG AND NAME **PS3: EVALUATION OF EFFECTIVENESS**

DESCRIPTION Analysis of efficiency of alternative development strategies and analysis of trade-offs.

TOPICS Minimum number of hours: 3

1. Use of linear programming
2. Use of benchmarking
3. Use of cost benefit analysis

PREREQUISITES PS2

REQUISITE FOR AD3

SUBJECT NAME **PROBLEM SOLUTION METHODOLOGIES**

KNOWLEDGE UNIT TAG AND NAME **PS4: OVERVIEW OF REQUIREMENTS SPECIFICATION**

DESCRIPTION Overview of the development of specifications for defining software requirements.

TOPICS Minimum number of hours: 1
1. Requirements analysis
2. Informal specifications
3. Formal specifications

PREREQUISITES PS2

REQUISITE FOR AD1

SUBJECT NAME **PROBLEM SOLUTION METHODOLOGIES**

KNOWLEDGE UNIT TAG AND NAME **PS5: VERIFICATION AND VALIDATION**

DESCRIPTION Methodologies to assure production of quality software.

TOPICS Minimum number of hours: 3
1. Pre-conditions and post-conditions
2. Invariants
3. Testing strategies and verification of program correctness
4. Tracing
5. Test data

PREREQUISITES PS2

REQUISITE FOR AD4

SUBJECT NAME **PROBLEM SOLUTION METHODOLOGIES**

KNOWLEDGE UNIT TAG AND NAME **PS6: DOCUMENTATION**

DESCRIPTION General requirements for documentation.

TOPICS Minimum number of hours: 2
1. Types of documentation
2. Content
3. Style

PREREQUISITES PR1, PS2

REQUISITE FOR None

SUBJECT NAME **PROBLEM SOLUTION METHODOLOGIES**

KNOWLEDGE UNIT NAME AND TAG **PS7: INTRODUCTION TO QUALITATIVE METHODS**

DESCRIPTION Strategies and methods other than mathematics where comparative elements are used to develop non-deterministic solutions to problems. The student should have an awareness of the range of solutions and knowledge that some solutions are beyond quantification (although they may have quantifiable elements).

TOPICS Minimum number of hours: 1
1. Decision analysis
2. Certainty and uncertainty
3. Risk analysis
4. Fit to the enterprise information system plan.

PREREQUISITES PS2

REQUISITE FOR AD3

SUBJECT NAME **PROBLEM SOLUTION METHODOLOGIES**

KNOWLEDGE UNIT TAG AND NAME **PS8: DECISION ANALYSIS**

DESCRIPTION Within organizations, there is a need to choose a course of action, to establish the criteria for selection or to assess a recommendation presented for approval. Decisions need to be based on thorough analysis of relevant information.

TOPICS Minimum number of hours: 2

1. Stating decisions
2. Establish and classify objectives
3. Evaluate alternatives
4. Assess the risks

PREREQUISITES AD2

REQUISITE FOR AD3

SUBJECT NAME **PROBLEM SOLUTION METHODOLOGIES**

KNOWLEDGE UNIT TAG AND NAME **PS9: PROBLEM ANALYSIS IN COMPUTER SYSTEMS**

DESCRIPTION Discussion of the complete problem solving process and analysis of a problem from defining to finding and verifying true cause.

TOPICS Minimum number of hours: 4

1. Finding the deviation
2. Specify the problem
3. Develop possible causes
4. True cause (Does it explain the *is* or *is not*?)
5. Verify the cause

PREREQUISITES PS5

REQUISITE AG2

SUBJECT NAME **PROBLEM SOLUTION METHODOLOGIES**

KNOWLEDGE UNIT TAG AND NAME **PS10: POTENTIAL PROBLEM ANALYSIS**

DESCRIPTION Once decisions are made to pursue a course of action the decision must be implemented. Usually the plan is implemented in a very dynamic environment. Potential problem analysis provides a conscious approach to help improve planning and give greater assurance of successful implementation.

TOPICS Minimum number of hours: 4

1. Anticipate potential problems
2. Anticipate likely causes
3. Selecting preventive actions
4. Select contingent actions

PREREQUISITES PS8

REQUISITE FOR AG2

SUBJECT NAME **PROBLEM SOLUTION METHODOLOGIES**

KNOWLEDGE UNIT TAG AND NAME **PS11: PROJECT MANAGEMENT**

DESCRIPTION Awareness of project management to organize and control a project.

TOPICS Minimum number of hours: 1

1. Definition of project purpose
2. Project planning tools - work breakdown structure (WBS), Gantt or bar charts, and critical path methods (CPM)
3. Project monitoring and control

PREREQUISITES AD2

REQUISITE FOR None

SUBJECT NAME **SOCIAL, ETHICAL, AND PROFESSIONAL ISSUES**

KNOWLEDGE UNIT TAG AND NAME **SP1: EVOLUTION OF COMPUTING AND THE PROFESSIONAL**

DESCRIPTION History of the computing profession culminating with the definition of the discipline and the role of today's computing professional.

TOPICS Minimum number of hours: 1

1. Evolution of computing as a discipline in its own right
2. The role of the professional
3. Codes of ethics and good conduct
4. Handling of violations

PREREQUISITES None

REQUISITE FOR None

SUBJECT NAME **SOCIAL, ETHICAL, AND PROFESSIONAL ISSUES**

KNOWLEDGE UNIT TAG AND NAME **SP2: INTELLECTUAL PROPERTY**

DESCRIPTION Introduction to the concept of software and algorithms as intellectual property, and means of legally protecting that property.

TOPICS Minimum number of hours: 2

1. Definition of intellectual property
2. Copyright and patent protection
3. Protection afforded to trade secrets

PREREQUISITES None

REQUISITE FOR None

SUBJECT NAME **SOCIAL, ETHICAL, AND PROFESSIONAL ISSUES**

KNOWLEDGE UNIT TAG AND NAME **SP3: SOFTWARE PROTECTION AND SECURITY**

DESCRIPTION Survey of design methodologies to reduce the incidence of software theft and misuse.

TOPICS Minimum number of hours: 2
1. Software ownership and license
2. Software copy protection and encryption
3. Hardware keys
4. Protection of data storage and transmission

PREREQUISITES None

REQUISITE FOR None

SUBJECT NAME **SOCIAL, ETHICAL, AND PROFESSIONAL ISSUES**

KNOWLEDGE UNIT TAG AND NAME **SP4: SYSTEM SECURITY**

DESCRIPTION Survey of measures to protect computer systems and databases from unauthorized use and tampering.

TOPICS Minimum number of hours: 2
1. Disaster planning
2. Physical security
3. Virus detection and prevention
4. Employee practices and standards.

PREREQUISITES None

REQUISITE FOR None

SUBJECT NAME **SOCIAL, ETHICAL, AND PROFESSIONAL ISSUES**

KNOWLEDGE UNIT TAG AND NAME **SP5: SOCIAL RESPONSIBILITY OF PROFESSIONALS**

DESCRIPTION The computing specialist as a professional with responsibilities to society.

TOPICS Minimum number of hours: 2

1. Legal and social consequences of errors in software
2. Professional associations
3. Licensing and certification of computing professionals
4. Liability issues
5. Professional relationships with clients

PREREQUISITES None

REQUISITE FOR None

SUBJECT NAME **SOCIAL, ETHICAL, AND PROFESSIONAL ISSUES**

KNOWLEDGE UNIT TAG AND NAME **SP6: DATA COLLECTION AND PRIVACY**

DESCRIPTION An introduction to the problems surrounding fair use of data, the right of individuals to privacy, and reporting requirements.

TOPICS Minimum number of hours: 1

1. Requirements for protecting the privacy of individuals
2. Data reporting under right-to-know laws
3. Record matching and its implications

PREREQUISITES None

REQUISITE FOR None

SUBJECT NAME **SOCIAL, ETHICAL, AND PROFESSIONAL ISSUES**

KNOWLEDGE UNIT TAG AND NAME **SP7: RISKS IN LARGE SYSTEMS**

DESCRIPTION Survey of issues that accompany the development of large software systems.

TOPICS Minimum number of hours: 1

1. Design and implementation teams
2. Software validation and testing
3. Risk assessment
4. Liability issues in large systems.

PREREQUISITES None

REQUISITE FOR None

PART III

CURRICULA IMPLEMENTATION SAMPLES

1.0 CURRICULA OVERVIEW

There are two sample curricula presented in this report. One is designed to prepare students to transfer to a baccalaureate degree in computer information systems, information systems, or management information systems; the other is intended to prepare students for entry-level positions in mainframe, mid-range or client/server, or small systems environments. Sample courses to support each of these curricula are also included.

1.1 Course Structure

The following components are included in each course description:

Course Title

A clear but brief name of the course with course code or number is provided.

Number of Semester Hours

The total number of semester credits and hours are listed in the following format:

Lecture (number of credits : number of hours)
Laboratory (number of credits : number of hours)

Prerequisites

Courses are listed that are to be completed prior to taking the course being described.

Goal or Purpose of the Course

A short paragraph expresses what the course should accomplish.

Behavioral Objectives for Students

Competencies expected from students who successfully complete the

course are detailed.

CIP2 Introduction to Programming for Information Processing Lecture (2:2)
Laboratory (1:2)

Prerequisite:
Computer literacy or equivalent; high school algebra
Knowledge of programming desirable

Goal:
Introduction to problem solving strategies, algorithm development and verification. The course emphasizes problem solving methods, algorithm design, and implementation and testing solutions using a procedural programming language in a business context.

Objectives:
Upon successful completion of this course, the student will be able to:

- Break down a simple problem into solvable components
- Apply standard design techniques to developing a solution
- Validate the solution by use of a test plan
- Implement using a procedural programming language
- Understand and use of appropriate simple data structures: arrays, pointers.

Subject Matter:

KU Tag	Portion of Hours		Depth	Emphasis
AD4	6/14	3	T,A,D	
AD5	1/13	2	A,D	
AD7	1/6	1	T	
AG4	1/9	2	T	
CF4	1/2	2	T	
CF10	1/3	2	T	
DB1	1/4	2	T,A	
DB3	1/4	2	T,A	
DB4	1/2	2	T,A,D	
PR2	4/7	4	T,A,D	
PR3	3/4	4	T,A,D	
PR4	3/4	4	T,A,D	
PR5	2/4	4	T,A,D	
PR6	1/1	3	T,A,D	
PR7	2/4	3	T,A,D	
PR8	1/1	1	T	
PR9	1/1	1	T	

PR10	2/2	1	T
PS2	2/6	3	A,D
PS3	1/3	2	T,A,D
PS4	1/1	1	T,A
PS5	2/3	3	T,A,D
PS6	1/2	2	A,D

Closed Laboratories:

- introductory knowledge of design/problem solving techniques to simple, structured, well-defined problems
- create a test plan and test effectiveness of proposed solution
- introductory knowledge of a block-structured, procedural programming language
- application of simple data structures to appropriate problems
- addition of specified components to a program and verification of correct execution.

Figure III-1 Course Format Example

Subject Matter

A table is constructed with the following four columns:

(a) Knowledge Unit Tag

(b) Fraction x/n, containing the number of hours, x, of the knowledge unit in this course to the total number of hours, n, required for the entire knowledge unit.

(c) Depth of knowledge expected to be achieved for the knowledge unit within this course.

(d) Emphasis of knowledge to be achieved.

Laboratory/Exercise Component

Suggestions for open or closed laboratory experiences or exercises for the subject matter are given together with a description of any equipment or materials needed. The laboratory component is an essential experience and should be included in all required courses within a curriculum.

Figure III-1 contains a example of the format for course descriptions.

1.2 Emphasis of Knowledge

Within courses, each knowledge unit contains a listing of its emphasis. The emphasis is indicated by some combination of the three letters T, A, or D, which stand for the three paradigms of theory, analysis or design, respectively. Theory, which is deeply rooted in mathematics, corresponds to the fundamental knowledge base of the particular unit that is generally supported by fact or accepted understanding. To develop a coherent, valid theory, one uses the mathematical concepts of definition, theorem, and proof. Analysis, or modeling, corresponds to the scientific method used in verifying or testing a theoretical phenomenon, or in discovering a certain principle in the subject. With this paradigm, one forms a hypothesis, constructs a model and makes predictions, designs an experiment and collects data, and finally analyses the results. Design, which is rooted in engineering, corresponds to building an entity that demonstrates the general theme of the knowledge unit. With this paradigm, one states the requirements, states the specifications, designs and implements the system, and then tests the results. For example, consider the knowledge unit:

PR3: SEQUENCE CONTROL

For this knowledge unit, depending on the scope of the given course, the student might be expected to understand the language elements in evaluating an expression (theory emphasis), determine the effects of different expression

evaluation in a laboratory setting (analysis emphasis), and/or implement an algorithm by writing a program (design emphasis).

1.3 Depth of Knowledge

Within courses, the knowledge units contain a listing of the expected depth of knowledge or understanding of the knowledge unit. This indicator is identified by an integer 1 through 5. The interpretation of these integers is:

1 Awareness as expressed by general knowledge, definition, or recognition.

2 Description as evidence of conceptual understanding.

3 Differentiation or understanding as expressed by the ability to compare/contrast or to make connections with related topics.

4 Application or use as evidence of using predefined principles, methods, or tools in a well-defined environment.

5 Judgment as evidence of innovative decision making based upon analysis, synthesis or evaluation.

The depth of knowledge or understanding as described above, is considered minimal for each knowledge unit within its course setting. A depth of 5 is probably beyond the scope of courses in a two-year degree program.

1.4 Curriculum Structure

The following components are included to describe each curricula:

Curriculum Title and Degree
A specification is provided of the degree or certificate which will be granted when all requirements are fulfilled.

Purpose
A statement of the purpose for each curriculum is provided, followed by the purpose of each option within the curriculum (if any).

Curriculum Course Content
This section includes the list of courses that are required in the

curriculum, including the different options (when applicable), possible electives, courses from other disciplines, and general education requirements.

Course Descriptions
Detailed course descriptions are presented in Section 4.

Sample Curriculum Plan
For each sample, a sequence of core courses is given. For the career samples, a semester course plan is also given.

Exit Competencies
Students completing a curriculum in computing for information processing are expected to demonstrate certain competencies after completing the curriculum.

1.5 Curriculum Courses

The required core courses for the career and the transfer computing for information processing curricula are:

CIP 1 Introduction to Computing for Information Processing
An introduction to information processing, this course covers the role of computing and its impact on the business organization. Introduces the use of productivity tools to process information and develop applications.

CIP 2 Introduction to Procedural Programming in Information Processing
This course covers problem solving strategies, algorithm development and verification, includes use of a procedural programming language.

CIP 3 Advanced Information Processing for Business
This course covers file processing and database in the context of transaction processing systems.

CIP 4 Information Processing Systems Development I
This is the first of a two semester sequence leading students through the entire systems development life cycle, including experience of maintenance.

CIP 5 Information Processing Systems Development II
This is the second of a two semester sequence leading students through the entire systems development life cycle, including experience of maintenance.

CIP 6 Software Package Evaluation and Package-based Systems Development

This course will demonstrate the ability to select and implement package-based systems in a business environment.

The career computing for information processing curriculum requires two additional courses to be chosen, with constraints, from the following list.

CIP 7 Applications Project

This is a realistic project experience using knowledge gained from previous courses.

CIP 8 Current Issues in Information Technology

This course introduces students to the investigation and evaluation of current computing issues in the local business community.

CIP 9 Software Engineering

This course is a detailed treatment of software engineering practices, including CASE technology.

CIP 10 Advanced Languages

This course is an introduction to emerging programming technologies and specialty languages.

CIP 11 Prototyping

This course is a detailed treatment of the use of prototyping and the changing role of the programmer analyst.

CIP 12 Expert Systems

This is a course dealing with expert systems concepts, languages, shells, and applications.

CIP 13 Relational Databases

This course is a detailed treatment of relational database technology and its applications..

CIP 14 Object-Oriented Programming

This course is an introduction to the use of object-oriented programming in business applications.

CIP 15 Distributed Processing

In this course, students will develop an understanding of current networking technology as it applies to information processing.

CIP 16 Graphical User Interface Applications Development

This course prepares students to use a graphical user interface (GUI) effectively in the development of information processing applications.

CIP 17 Introduction to Multimedia

This course involves the design of educational software for microcomputers, including computer tutorials and interactive video disk applications.

Other courses in each curriculum should be chosen to be relevant to the students' requirements. In addition, there may be local restrictions and requirements imposed by the institution or governing body. However, the emphasis in the courses should be on application in the business environment rather than theory only. It is particularly important that the English requirements in the career program include communication and writing in a business setting. Supporting courses should be selected from each of the following categories. Some suggested courses are included in each category.

Business: Accounting, finance, management, and business law.

Mathematics: Discrete mathematics, statistics, calculus for business, and calculus.

Science: Laboratory-based sciences such as physics, chemistry, and biology.

Humanities / Social Science: Philosophy (logic, ethics), economics, political science, psychology, and sociology.

English / Communications: Technical writing, writing for documentation, speech, writing for publication, and presentation making and interviewing techniques.

2.0 UNDERLYING THEMES

The courses are designed to build knowledge and skills through sequencing and

repetition of application of knowledge to more complex environments. That is, one course may introduce concepts, skills, tools, techniques, at an elementary level; a subsequent course uses that knowledge and applies it in a more realistic, less structured environment. At the same time the general education component complements the technical curriculum by putting the subject matter into a broader context and relating it to the business world.

Certain fundamental concepts within the discipline play an important role in the design of individual courses and the curriculum. These underlying themes represent a *glue* that can be used to provide cohesiveness, a view towards how all these elements work together. They represent fundamental abstractions, principles, technical problems, and issues that a student completing this program will encounter in a variety of circumstances.

Curriculum topics can convey a wholeness by acknowledging and discussing these underlying themes as they appear during the students' educational experiences. Done properly, the following benefits can occur: Minimize the perception that the discipline is a fragmented collection of unrelated topics. Since the underlying themes span many of the subject areas, they serve to unify the discipline. Learning will be facilitated by the presence of generalizations and analogies. Given a reference to an already understood concept, students can assimilate new ideas more easily.

Below is a list of the underlying themes for the discipline of computing for information processing:

Binding: The process of making abstraction more concrete by assigning properties to it. Some examples are associating a type with a variable name, associating a process with a processor, and associating a specific task with reference manuals.

Complexity of large problems: The effects of the nonlinear increase in complexity as the size of a problem grows. This is an important factor in distinguishing and selecting methods to solve an organizational problem/task.

Conceptual models: Ways of characterizing, visualizing, and thinking about an idea or problem.

Control: How hardware/software work together to effect a predictable desired result.

Design concepts: How things work as a result of design versus how it is

installed/configured - or what happens as a result of an update or change.

Effectiveness: The extent to which a system is complete and correct. That is, the extent to which a system satisfies its specifications.

Efficiency: A measure of the resources required for the execution of a process, the extent to which a system uses minimal resources.

Evolution: The changing nature of requirements that causes essentially continuous revision to a software or hardware implementation.

Formal models: Concepts, notations, and abstractions.

Levels of abstraction: The ability to represent an entity by abstractions having different levels of detail and specificity. This provides the ability to use a model having only the minimum level of detail needed, thus providing precision and generality.

Management: The supervision of the system or the software.

Ordering in space: The relative position of entities in a physical or virtual machine or system. This applies for instance, to communication paths in a local area network.

Ordering in time: The temporal aspects that surround the execution of a command, as determined by control information and the execution state. Examples are the concurrency controls in a database application.

Problem resolution: The systematic collection of information relating to a problem and the ability to synthesize to form relevant data to solve and verify a problem.

Reliability: The extent to which systems behave according to specifications under all conditions. This includes responses to error conditions as well as graceful handling of unanticipated situations.

Reuse: Realizing the potential for components of one system to be used in another system. This applies to using parts from one computer to another, to using one component in an application for another system.

Security: The extent to which hardware and software systems are vulnerable to misuse, as well as effective methods for defending against misuse.

Data and system restoration after a calamity is another dimension of security.

Standards: Application and implementation of standards, either industry-wide or company-specific. An example is the installation and use of a software package; it is necessary to understand the company's standards on program function keys in order to configure the software.

Tradeoffs and consequences: The economic, cultural, and other effects of selecting one design alternative over another. Examples are the consequences of selecting between one software package and another, or one hardware system and another.

Underlying themes are independent of any particular discipline and are more fundamental than any occurrence of their use. These ideas, principles, and processes help to unify the program and must be communicated throughout each curriculum.

3.0 SAMPLE CURRICULA

This section includes sample curricula for the transfer and the career two-year associate degree programs in computing for information processing.

3.1 Sample Curriculum for Transfer Students

Curriculum Title and Degree
Title: Computing for Information Processing
Degree: Associate

Purpose
This program of study is designed to prepare students to transfer, with full junior-level status, to a baccalaureate degree program in computer information systems, information systems, and management information systems. Since it is up to the transfer institution to determine what courses will satisfy the requirements of their first two years, and since many four-year institutions offer most of their computing courses at the junior and senior years, the following suggested program of study should be carefully coordinated with the recommendations and the requirements of the transfer institutions.

Curriculum Course Content
Required Computing Courses
CIP 1 Introduction to Computing for Information Processing
CIP 2 Introduction to Procedural Programming in Information Processing
CIP 3 Advanced Information Processing for Business
CIP 4 Information Processing Systems Development I
CIP 5 Information Processing Systems Development II
CIP 6 Software Package Evaluation and Package-based Systems Development

Recommended Computing Courses
Other computing courses may be offered or required including any of the career-oriented courses CIP 7 through CIP 17, depending upon the recommendations of the transfer institution.

Other courses
The choice of other courses is strongly dependent on the requirements of the transfer institution. Students will need courses in English and communications, social science and humanities, mathematics, and science. Suggested courses include the following: accounting, finance, management, business communications, statistics, quantitative methods, discrete mathematics, ethics, and economics. Other course suggestions may be found in Section 1.5.

Sample Curriculum Plan

Table III-1 contains a sample four-semester plan of study for the transfer associate degree program in computing for information processing. Figure III-2 contains a diagram of the sequencing of the required computing courses. Table III-2 is a tabular listing by lecture hour of the subject area content which is contained in each of the sample required courses. Sample exit competencies for each of the knowledge units is included in Figure III-3. The descriptions of the courses CIP 1 through CIP 6 are included in Section 4.1.

Semester 1	**Credits**		**Semester 2**	**Credits**
CIP 1	4		CIP 3	4
CIP 2	3		CIP 4	4
Business 1	3		Business 2	3
English 1	3		English 2	3
Mathematics 1	3		Mathematics 2	3
Total Credits	16		Total Credits	17
Semester 3	**Credits**		**Semester 4**	**Credits**
CIP 5	4		Social Science 2	3
CIP 6	4		Science 2	4
Social Science 1	3		Elective 2	3
Science 1	3		Elective 3	3
Elective 1	3		Elective 4	3
Total Credits	17		Total Credits	16

Table III-1 Course Scheduling Sequence - Transfer Program

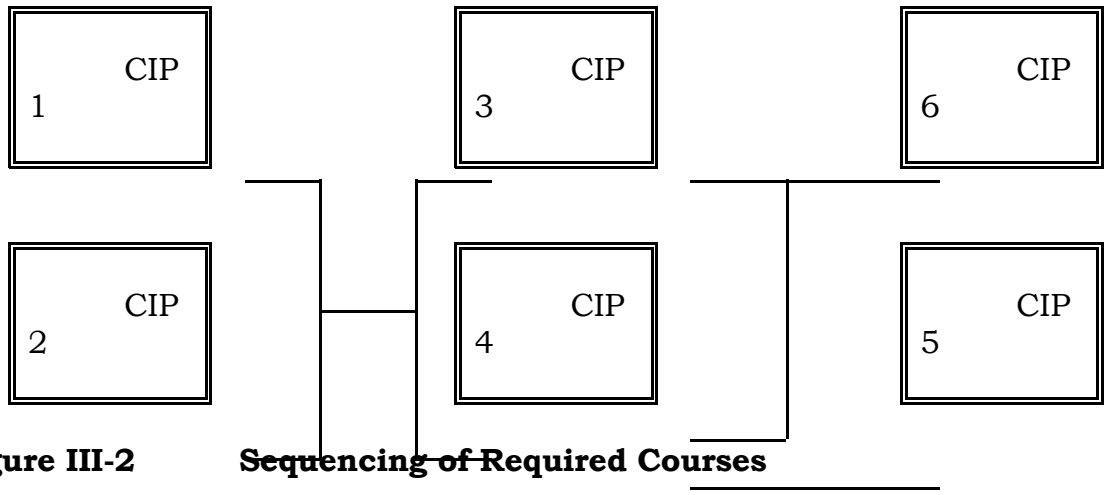

Figure III-2 Sequencing of Required Courses

Subject Name	CIP 1	CIP 2	CIP 3	CIP 4	CIP 5	CIP 6
Application Development Strategies (AD)	3	8	12	16	20	10
Application Generation Strategies (AG)	9	1	3	2	4	13
Computer Concepts and Fundamentals (CF)	15	2	3	0	2	5
Database and Information Retrieval (DB)	1	3	11	5	2	0
Interpersonal and Comm. Technique (IC)	1	0	0	8	4	2
Organizational Behavior (OB)	2	0	0	2	1	1
Programming (PR)	5	19	6	0	0	0
Problem Solution Methodologies (PS)	2	7	4	3	7	5
Social, Ethical, and Professional Issues (SP)	2	0	1	4	0	4
TOTALS	40	40	40	40	40	40

Table III-2 Content Hours of Required Courses

3.2 Sample Curriculum for Career Students

Curriculum Title and Degree
Title: Computing for Information Processing
Degree: Associate

Purpose
This program of study is designed to prepare students for entry-level positions in business, industry, and government. The choice of tools and techniques to implement the program is a local decision, since it should be representative of the local business community. The courses will give students experience in the broad spectrum of computing environments in use so that, upon graduation, their career choices are not limited by early specialization.

Curriculum Course Content
Required Computing Courses
CIP 1 Introduction to Computing for Information Processing
CIP 2 Introduction to Procedural Programming in Information Processing
CIP 3 Advanced Information Processing for Business
CIP 4 Information Processing Systems Development I
CIP 5 Information Processing Systems Development II
CIP 6 Software Package Evaluation and Package-based Systems Development
CIP A (Required Elective)
CIP B (Required Elective)

CIP A and CIP B are two additional required computing courses to be chosen from the career-oriented courses CIP 7 through CIP 17. The choice options should be contingent upon the particular emphasis of the career program. Sample emphases are listed in Section 3.3. The local institution must determine the emphasis of their career program. This emphasis should be dependent upon the needs of the local business and industry, the expertise of the faculty, and the availability of specialized equipment and software. Additional computing courses may be offered or required, as needed and could be chosen from other programs such as CSS. Appropriate courses may be chosen as electives.

Other courses
Students should be required to take two courses in each of the categories of business, mathematics, science, humanities/social science, and English/ communications, as listed in Section 1.5 of Part III. Additional courses may be necessary, depending upon the general education requirements of the local institution.

Sample Curriculum Plan

Table III-3 contains a sample four-semester plan of study for the career associate degree program in computing for information processing. Figure III-2 contains a diagram of the sequencing of the required computing courses CIP 1 through CIP 6. Table III-2 is a tabular listing by lecture hour of the subject area content contained in each of the sample required courses CIP 1 through CIP 6. Sample exit competencies for each of the knowledge units is included in Figure III-3. The descriptions of the courses CIP 1 through CIP 17 are included in Section 4.

Semester 1	**Credits**		**Semester 2**	**Credits**
CIP 1	4		CIP 3	4
CIP 2	3		CIP 4	4
Business 1	3		Business 2	3
English 1	3		English 2	3
Mathematics 1	3		Mathematics 2	3
Total Credits	16		Total Credits	17
Semester 3	**Credits**		**Semester 4**	**Credits**
CIP 5	4		CIP A	3
CIP 6	4		CIP B	4
Social Science 1	3		Social Science 2	3
Science 1	3		Science 2	3
Elective 1	3		Elective 2	3
Total Credits	17		Total Credits	16

Table III-3 Course Scheduling Sequence - Career Program

3.3 Sample Emphases for Career Curriculum

The following are sample emphases for the career computing for information processing associate degree program, along with suggested choices for the two required computing electives CIP A and CIP B:

Host-Based (Procedural Systems) Emphasis
CIP 7 and one course from the list: CIP 8, CIP 9, CIP 10, CIP 11, CIP 13.

Host-Based (DBMS/4GL) Emphasis
Two courses from the list: CIP 7, CIP 8, CIP 9, CIP 11, CIP 12, CIP 13, CIP 14.

Client/Server Emphasis
Two courses from the list: CIP 7, CIP 8, CIP 9, CIP 10, CIP 11, CIP 13, CIP 14

Microcomputer Systems Emphasis
Two courses from the list: CIP 10, CIP 11, CIP 13, CIP 15, CIP 16, CIP 17.

4.0 DESCRIPTIONS OF SAMPLE COURSES

This section contains descriptions of all courses that support the transfer and the career computing for information processing associate degree programs. The descriptions of the courses CIP 1 through CIP 6 are listed in Section 4.1, following the format discussed in Section 1. The descriptions of the courses CIP 7 through CIP 17 are listed in an abbreviated form in Section 4.2.

4.1 Courses Required for Transfer and Career Students

CIP 1 and CIP 2 have a prerequisite of computer literacy. For the prerequisite, computer literacy means that the student is expected to have a working knowledge of the use of the computer and related devices. The knowledge must include basic use of common productivity software such as word processing, spreadsheets, and databases. The student must also have a working knowledge of the system commands needed to be functional with the computer. This prerequisite could be satisfied with the completion of the course COD 1 - Computer Communications, which is fully described in the Computing for Other

Disciplines Committee report. In CIP 2, a prior knowledge of programming is desirable. This programming background could be satisfied with the completion of the course CS 100 - Introduction to Programming, which is fully described in the Computing Sciences Committee report.

CIP 1 Introduction to Computing for Information Processing Lecture (3:3)
Laboratory (1:2)

Prerequisite:

Computer literacy or equivalent

Goal:

Understanding of the variety of configurations, and the capabilities and limitations of computer-based systems in information processing. Use of productivity tools to develop applications.

Objectives:

Upon successful completion of this course, the student will be able to:

- Understand the broad range of computing systems available and their applications in information processing
- Understand the impact of computing applications on the business organization and society
- Describe the function of information processing in an organization
- Understand the interdependence of hardware and software
- Understand the functions of an operating system and relate them to single user and multi-user operating environments
- Describe the functions of operating systems with particular reference to networking products common in the local employment community
- Be aware of the functions of local and wide area networks
- Understand the functions and operation of file servers, communication packages, file transfer software
- Describe the application of data communication to information systems processing.
- Apply software tools, such as word processing and spreadsheets, to develop simple applications with custom menus, screens, and reports
- Apply a database package to perform record and file manipulations for creating and maintaining data, including report generation and menus

Subject Matter:

KU Tag	Portion of Hours	Depth	Emphasis
AD5	3/13	2	T,A,D
AG1	7/10	3	T,A,D
AG3	1/5	2	T,A,D
AG4	1/9	1	T
CF1	1/1	1	T
CF2	1/2	1	T
CF3	1/1	2	T
CF4	1/2	1	T
CF5	2/2	2	T
CF6	1/1	2	T
CF7	1/1	2	T
CF8	1/1	2	T
CF9	1/3	1	T
CF10	1/3	1	T
CF11	1/3	1	T
CF12	1/3	1	T
CF13	1/1	1	T
CF14	1/3	1	T
DB2	1/1	3	T
IC3	1/2	2	T
OB1	1/3	1	T
OB4	1/1	1	T
PR1	1/1	1	T
PR2	3/7	3	T,A,D
PR3	1/4	2	T,A
PS2	2/6	2	T,A,D
SP1	1/1	2	T
SP5	1/2	2	T

Closed Laboratories:

- Use of a spreadsheet including creating macros, menus, and programs
- Use of a database package to create and modify data, produce reports, and create programs and menus
- Installation and setup of common productivity software in a single user environment
- Simple backup and recovery procedures in a single user environment
- Observe use of communication package(s)
- Ability to navigate through a set of windows using a mouse.

CIP 2 Introduction to Programming for Information Processing Lecture (2:2)
Laboratory (1:2)

Prerequisite:

Computer literacy; high school algebra
Knowledge of programming desirable

Goal:

Introduction to problem solving strategies, algorithm development and verification. The course emphasizes problem solving methods, algorithm design, and implementation and testing solutions using a procedural programming language in a business context.

Objectives:

Upon successful completion of this course, the student will be able to:

- Break down a simple problem into solvable components
- Apply standard design techniques to developing a solution
- Validate the solution by use of a test plan
- Implement using a procedural programming language
- Select, understand and apply appropriate simple data structures including arrays and pointers
- Understand and use standard algorithms such as sorting and searching
- Be aware of the impact of maintenance activity on program execution

Subject Matter:

KU Tag	Portion of Hours	Depth	Emphasis
AD4	6/14	3	T,A,D
AD5	1/13	2	A,D
AD7	1/6	1	T
AG4	1/9	2	T
CF4	1/2	2	T
CF10	1/3	2	T
DB1	1/4	2	T,A
DB3	1/4	2	T,A
DB4	1/2	2	T,A,D
PR2	4/7	4	T,A,D
PR3	3/4	4	T,A,D
PR4	3/4	4	T,A,D
PR5	2/4	4	T,A,D
PR6	1/1	3	T,A,D
PR7	2/4	3	T,A,D
PR8	1/1	1	T
PR9	1/1	1	T
PR10	2/2	1	T
PS2	2/6	3	A,D
PS3	1/3	2	T,A,D
PS4	1/1	1	T,A
PS5	2/3	3	T,A,D
PS6	1/2	2	A,D

Closed Laboratories:

- Introductory knowledge of design/problem solving techniques to simple, structured, well-defined problems
- Create a test plan and test effectiveness of proposed solution
- Introductory knowledge of a block-structured, procedural programming language
- Application of simple data structures to appropriate problems
- Addition of specified components to a program and verification of correct execution.

CIP 3 Advanced Information Processing for Business Lecture (3:3)
Laboratory (1:2)

Prerequisite:

CIP 1 and CIP 2

Goal:

Knowledge of file processing and database management in the context of transaction processing systems.

Objectives:

Upon successful completion of this course, the student will be able to:

- Use advanced features of a procedural business-oriented language
- Implement a simple system for transaction processing involving validation, file processing and reporting
- Understand and implement report programs involving sorting, selection, editing, and summarization of data
- Create and update sequential and indexed sequential files
- Select appropriate file organization for a solution
- Identify and use the principal components of a database system
- Create and update database files.

Subject Matter:

KU Tag	Portion of Hours	Depth	Emphasis
AD1	2/5	3	T,A,D
AD2	1/2	2	T,A
AD4	3/14	3	A,D
AD5	1/13	3	A,D
AD6	1/4	2	T,A
AD7	1/6	2	T,A
AD10	3/4	3	T,A
AG3	1/5	2	T,A
AG4	2/9	3	T,A
CF9	2/3	3	A
CF10	1/3	3	A
DB1	3/4	2	T,A,D
DB3	3/4	4	T,A,D
DB4	1/2	3	T,A,D
DB5	2/4	2	T,A,D
DB6	2/5	2	T,A,D
PR4	1/4	4	T,A,D
PR5	2/4	4	T,A,D
PR7	2/4	3	T,A,D
PR11	1/1	1	T
PS2	2/6	4	T,A,D
PS3	1/3	3	T,A,D
PS5	1/3	4	T,A,D
SP6	1/1	1	T

Closed Laboratories:

- Transaction creation and validation
- Sequential file processing
- Report programming
- Indexed sequential file processing
- Elementary database processing.

CIP 4 Information Processing Systems Development I Lecture (3:3)
Laboratory (1:2)

Prerequisite:

CIP 1 and CIP 2

Goal:

The first course in a two-semester sequence leading students through the entire systems development life cycle of a realistic, well-defined system. Topics include analysis, prototyping, and documentation.

Objectives:

Upon successful completion of this course, the student will be able to:

- Understand the systems development life cycle and be aware of different development strategies
- Capture user requirements
- Document and present a systems proposal, including evaluation of alternative solutions and recommendation of preferred solution
- Be aware of the tradeoffs and consensus involved in the selection of a solution
- Know how to use prototyping / modelling to clarify requirements in the analysis phase
- Know how to use data modelling and conceptual design for files and databases to support the prototyping
- Prepare and present a walkthrough or review of the project's work
- Be aware of the impact of object-oriented programming on the systems development process
- Be aware of the involvement of end users in the remaining life cycle of the project.

Subject Matter:

KU Tag	Portion of Hours	Depth	Emphasis
AD1	3/5	3	A,D
AD2	1/2	2	A
AD3	9/10	3	T,A,D
AD7	1/6	3	A,D
AD8	1/8	2	T,A
AD9	1/3	2	T,A
AG3	1/5	3	A
AG5	1/1	2	T,A
DB5	2/4	3	A,D
DB6	3/5	3	A,D
IC1	1/1	2	T,A
IC2	3/4	3	T,A,D
IC4	1/2	3	D
IC5	3/6	2	T,A
OB1	1/3	3	T,A
OB2	1/1	2	T,A
PS1	1/1	2	T,A
PS7	1/1	2	T,A
PS8	1/2	2	T,A
SP2	2/2	2	T
SP5	1/2	2	T
SP7	1/1	2	T

Open and Closed Laboratories:

- Interviewing to gather user requirements
- Writing acceptable business communications
- Presentation workshops
- Model screens, reports, and menus to validate requirements

CIP 5 Information Processing Systems Development II Lecture (6:6)
Laboratory (2:4)

Prerequisite:

CIP 4

Goal:

The second course in a two-semester sequence leading students through the entire systems development life cycle of a realistic well-defined system. Topics include design, implementation, enhancement maintenance, backup, recovery, and documentation.

Objectives:

Upon successful completion of this course, the student will be able to:

- Know how to use prototyping / modelling in the design phase
- Design a solution using accepted design techniques, including screens, reports, programs, interfaces, and security
- Know how to use data modelling and conceptual design for files and databases
- Write/develop a system of programs using appropriate software, eg. application generators / database / 4GLs / CASE
- Convert files and implement backup and recovery procedures
- Write documentation (system and user)
- Perform simple enhancement maintenance of the system after installation
- Be aware of impact of project management techniques to organize and control the activities of the project.

Subject Matter:

KU Tag	Portion of Hours	Depth	Emphasis
AD4	3/14	3	A,D
AD5	5/13	3	A,D
AD6	3/4	3	A,D
AD7	3/6	3	A,D
AD8	3/8	3	A,D
AD9	2/3	3	A,D
AD10	1/4	3	A,D
AG3	1/5	3	A,D
AG4	2/9	3	A,D
AG6	1/1	2	A,D
CF11	1/3	2	T
CF14	1/3	2	T
DB7	2/2	2	T,A
IC2	1/4	3	A,D
IC3	1/2	2	T,A
IC5	2/6	3	A,D
OB3	1/1	2	A,D
PS3	1/3	3	T,A
PS6	1/2	3	A,D
PS9	2/4	3	T,A,D
PS10	2/4	3	T,A,D
PS11	1/1	1	T

Open and Closed Laboratories:

- Writing acceptable business communications
- Presentation workshops
- Design programs, screens, reports, menus, files, databases to meet those requirements
- Installation/implementation of the solution
- Backup and recovery procedures
- Enhancement assignment to provide maintenance experience.

CIP 6 Software Package Evaluation and Package-based Systems Development Lecture (3:3) Laboratory (1:2)

Prerequisite:

CIP 3 and CIP 4

Goal:

Identification of commercial software which meet users' requirements, essential properties of a good software package, how to shortlist, benchmark, and select, preparation of cost/benefit analysis, alternatives of buying, renting or leasing, installation, setup, and customizing. The course could be exercise driven or case study driven. (Selection of platforms is beyond the expertise acquired in this program and would be achieved by a graduate of the curriculum recommendations of the Computer Support Services Committee. Courses in their curriculum recommendations might be appropriate electives for students interested in this area.)

Objectives:

Upon successful completion of this course, the student will be able to:

- Be aware of essential properties of a good software package
- Identify selection criteria
- Know how to shortlist, benchmark, and select
- Be aware of how to prepare RFI (Request For Information) and RFP (Request For Proposal)
- Know how to prepare a cost/benefit analysis
- Know how to select alternatives of buying, renting or leasing
- Install a microcomputer package and implement, including customizing of screens, reports, menus
- Implement interfaces and security
- Implement backup and recovery procedures
- Write documentation (system and user)
- Be aware of management issues in a LAN environment

Subject Matter:

KU Tag	Portion of Hours	Depth	Emphasis
AD3	1/10	4	A,D
AD4	2/14	4	A,D
AD5	3/13	4	A,D
AD8	4/8	4	A,D
AG1	3/10	4	A,D
AG2	6/6	4	T,A,D
AG3	1/5	4	A,D
AG4	3/9	4	A,D
CF2	1/2	2	T,A
CF11	1/3	2	T
CF12	2/3	3	A,D
CF14	1/3	2	A
IC4	1/2	3	D
IC5	1/6	3	A,D
OB1	1/3	3	T,A
PS8	1/2	3	A,D
PS9	2/4	3	A,D
PS10	2/4	3	A,D
SP3	2/2	3	T
SP4	2/2	2	T

Open and Closed Laboratories:

- Installation and setup of common productivity software, sharing files, in a networked environment
- Installation and setup of a specialized industry standard product (eg. accounting, inventory, project management), sharing files, in a networked environment
- Research in journals to evaluate industry-standard software packages to meet specified requirements
- Use productivity software to develop user menus or scripts
- Writing of interfaces or utility software
- Backup and recovery procedures
- Documentation (system and user).

Tag	Name	Depth
AD1:	Specification Methods	3
AD2:	Life Cycle	2
AD3:	Analysis	4
AD4:	Design	4
AD5:	Implementation	4
AD6:	Post Implementation Activities	3
AD7:	Quality Assurance, Security, Standards	3
AD8:	Prototyping	4
AD9:	Human/Machine Interfaces	4
AD10:	Use of System Queues	3
AG1:	Common Productivity Software	4
AG2:	Purchased Application Packages	4
AG3:	Fourth Generation Development Methodology	3
AG4:	Application Generation Software	3
AG5:	Front-End CASE Tools	2
AG6:	Back-End CASE Tools	2
CF1:	Historical Perspective	1
CF2:	Emerging Technologies	1
CF3:	Overview of Hardware	2
CF4:	Overview of Software	2
CF5:	Operating Systems	2
CF6:	Machine Organization	2
CF7:	Process Management	2
CF8:	Memory Management	2
CF9:	Auxiliary Storage	3
CF10:	Machine Representation of Data	3
CF11:	Introduction to Telecommunications	3
CF12:	Applications on Networks	3
CF13:	Security	3
CF14:	Systems Administration	3
DB1:	File and Physical Database Organization	2
DB2:	Introduction to Databases	3
DB3:	Database Models	3
DB4:	Logical Storage Structures	3
DB5:	Algorithms for Data Manipulations	4

DB6: Relational Models 3
DB7: Database Integrity 3

IC1: Presentation Skills 3
IC2: Technical Writing Skills 3
IC3: Technical Reading Skills 3
IC4: Electronic Communication Technologies 3
IC5: Teamwork 3

OB1: Uses of Information Technology 3
OB2: Organization Management Concepts 1
OB3: Management of Information Services 2
OB4: Regulatory Issues and Outside Influences 1

PR1: History of Programming Languages 1
PR2: Elements of a Programming Language 4
PR3: Sequence Control 4
PR4: Data Types 4
PR5: Structured Data Types 4
PR6: Introduction to Sorting and Searching 3
PR7: Data Access 3
PR8: Object-oriented Programming 1
PR9: Objects 1
PR10: Language Translation Systems 1
PR11: Programming Paradigms 1

PS1: Quantitative Methods 3
PS2: Algorithm Development 4
PS3: Evaluation of Effectiveness 3
PS4: Overview of Requirements Specification 1
PS5: Verification and Validation 4
PS6: Documentation 4
PS7: Introduction to Qualitative Methods 3
PS8: Decision Analysis 3
PS9: Problem Analysis in Computer Systems 3
PS10: Potential Problem Analysis 3
PS11: Project Management 1

SP1: Evolution of Computing and the Professional 1
SP2: Intellectual Property 2
SP3: Software Protection and Security 3
SP4: System Security 3
SP5: Social Responsibility of Professionals 3

SP6: Data Collection and Privacy 3
SP7: Risks in Large Systems 1

Figure III-3 Expected Exit Competencies

4.2 Elective Courses for Career Students

This section contains partial course descriptions, including goals and objectives, for the career program elective courses CIP 7 through CIP 17. The course descriptions in this section do not include knowledge units since the subject matter is not a part of the common body of knowledge required for all associate-degree graduates in computing for information processing. Additional courses may be chosen from other programs such as computer support services.

A student must choose two of the courses described in this section. The local institution must decide which of these courses are most appropriate for the emphasis of their career program. This emphasis should depend on the needs of the local industries, the expertise of the faculty, and the availability of specialized equipment and software. Example emphases were listed in Section 3.3.

Courses CIP 7 through CIP 17 are listed two per page in an abbreviated course format. The courses are not listed in any prioritized ordering.

CIP 7 Applications Project Lecture (1:1)
Laboratory (2:6)
Prerequisite:
CIP 5 and CIP 6

Goal:
Realistic project experience using tools and techniques from previous courses. Demonstrate application of knowledge in a less structured or less well-defined environment, within guidelines specified by faculty. Projects should be drawn from the local employment community and should require extensive contact with the business community. Students should work in teams and seek direction from a faculty member as project supervisor.

This is a capstone course which develops and demonstrates the students' ability to apply their knowledge. The instructional emphasis of the course is on case work, rather than formal classroom instruction. The scheduled class components will familiarize students with business management, organizational issues, and project management. This course is also an opportunity for students to use newer tools available in the business community. This may involve hands-on experience with, for example, CASE tools, object-oriented programming with C++, and expert systems.

Objectives:
Upon successful completion of this course, the student will be able to:

- Understand project management techniques
- Develop a work plan and schedule for a simple project
- Develop a system in a team environment including coordination and planning of activities with other team members and the faculty project manager
- Know when to seek guidance from a project manager in the less-structured environment
- Demonstrate application of prior knowledge in a less structured environment.

CIP 8 Current Issues in Information Technology Lecture (2:2)
Laboratory (1:2)

Prerequisite:
CIP 5

Goal:
Course to discuss and evaluate current issues in the local information technology community. Use of guest speakers, site visits, etc. would be appropriate. Topics may be wide-ranging and should reflect the local community. Suggested topics include: application generators, software engineering, project management, quality assurance, office systems, client/server architecture, emerging technologies, open systems and portability, expert systems, artificial intelligence, neural networks, object-oriented programming and object-oriented design. Students will be expected to consult journals in order to complete research projects on topics, evaluate material presented in the journals, and make presentations. The laboratory portion of the course could consist of student presentations and discussion of topics.

Objectives:
Upon successful completion of this course, the student will be able to:

- Be aware of current trends and issues in information technology
- Research a topic in industry journals
- Evaluate and summarize the material, distinguish between the value of information in journals and trade literature
- Present a paper or report
- Apply knowledge to simple case examples (which can be completed as an assignment).

CIP 9 Software Engineering Lecture (2:2)
Laboratory (1:2)

Prerequisite:
CIP 5

Goal:
A detailed treatment of software engineering practices and CASE technology. Course includes different methodologies, categories of CASE, software life cycle, reusability, code generation and implementation considerations, and the development of a major project.

Objectives:

Upon successful completion of this course, the student will be able to:

- Understand the impact of software engineering on the systems development life cycle
- Review and evaluate the tools available
- Understand how tools and techniques can be used to ensure software reliability, maintainability, reuse, and quality
- Understand the impact of implementation, integration, and maintenance
- Be aware of management issues
- Develop a project using CASE tools.

CIP 10 Advanced Languages (lecture (2:2)
Laboratory (1:2)

Prerequisite:
CIP 3

Goal:
An introduction to emerging programming technologies and specialty languages. The course includes: object-oriented constructs and use of a common object-oriented programming language such as C++ or Smalltalk; expert systems concepts and their uses in information processing; specialty languages such as statistical programming and simulation packages; and decision support systems.

Objectives:
Upon successful completion of this course, the student will be able to:

- Understand the concepts of objects, classes, methods, messages, and inheritance and apply them to a simple business problem
- Understand the impact of OOP on the design and construction of applications
- Understand the concepts of rule-based expert systems and inference processes
- Be aware of the variety of specialty languages and their applications
- Be able to apply a specialty language to an appropriate business problem
- Use a decision support tool to model a business problem and present alternative solutions

---

CIP 11 Prototyping Lecture (2:2)
Laboratory (1:2)

Prerequisite:
CIP 5

Goal:
A detailed treatment of the use of prototyping and the changing role of the programmer analyst. Includes the development and importance of a stable data model for the underlying architecture and the development of a major on-line project.

Objectives:

Upon successful completion of this course, the student will be able to:

- Understand the importance of a stable data model underlying systems development
- Be able to model data for a realistic business problem
- Use a 4GL (fourth generation language) to implement a realistic on-line application
- Know how prototyping affects the role of the programmer analyst and the user in the systems development process.

CIP 12 Expert Systems Lecture (2:2)
Laboratory (1:2)

Prerequisite:
CIP 5

Goal:
This course stresses expert systems concepts, languages and shells. Students will become proficient with a language such as LISP or Prolog for use as an expert system development tool. Students will also use a shell package to develop a project.

Objectives:
Upon successful completion of this course, the student will be able to:

- Understand and use a rule-based expert system
- Be aware of alternative expert systems paradigms
- Understand where expert systems can be used in information processes
- Use an expert system shell package to develop a simple project.

CIP 13 Relational Database Systems Lecture (2:2)
Laboratory (1:2)

Prerequisite:
CIP 5

Goal:
A detailed treatment of relational database technology. Topics include relational algebra and calculus, and its implementation in language, data modeling for realistic business applications, creation and use of views. Students will develop a major project.

Objectives:
Upon successful completion of this course, the student will be able to:

- Be aware of the mathematical foundations of the relational model
- Develop E-R diagrams for a realistic business problem

- Understand and use data normalization up to 4th normal form (4NF)
- Understand the need for views and the impact on data integrity
- Be aware of the differences among relational products on the market
- Use a relational product to develop a major product.

CIP 14 Object-Oriented Programming Lecture (2:2)
Laboratory (1:2)

Prerequisite:
CIP 3

Goals:
An introduction to the use of object-oriented programming in business applications. Topics include data abstraction, class hierarchies and inheritance, encapsulation, and message passing. Students will master a typical object-oriented programming language, use object-oriented extensions of popular languages, and develop a major project using these languages.

Objectives:
Upon successful completion of this course, the student will be able to:

- Understand the principles behind the paradigm
- Understand the concepts of objects, classes, methods, messages, inheritance and encapsulation
- Acquire experience with a typical Object-Oriented Programming Language to develop a realistic project
- Be aware of object-oriented extensions to existing languages and their impact on the development of a system.

CIP 15 Distributed Processing Lecture (2:2)
Laboratory (1:2)

Prerequisite:
CIP 6

Goal:
Students will develop an understanding of current networking technology as it applies to information processing. Topics include basic concepts and terminology related to data communications and networks, with particular emphasis on local-area networks. Students will develop a system and be able to create user menus, script files, use supervisor features, set performance requirements, and understand a network installation along with its associated management and security issues.

Objectives:

Upon successful completion of this course, the student will be able to:

- Understand the pros and cons of distributed systems and shared access
- Understand the concepts and terminology associated with data communication and networks
- Be able to use and evaluate micro to mainframe connections
- Be able to use configuration, shell and batch commands
- Be able to create user menus, script files and use supervisor features on a LAN (Local area network)
- Understand performance requirements.

CIP 16 Graphical User Interface Applications Development Lecture (2:2)
Laboratory (1:2)

Prerequisite:
CIP 3

Goal:
This course is an introduction to application development using a graphical user interface (GUI). Topics include: basic file management, use of desktop accessories, running application programs, printer and memory management, and customizing a graphical user interface.

Objectives:
Upon successful completion of this course, the student will be able to:

- Explain the advantages and disadvantages of a graphical user interface environment
- Manage files, databases, and directories using a graphical user interface
- Develop and run applications using a graphical user interface
- Associate documents with specific applications
- Print files with a print manager
- Understand memory management concepts
- Change the desktop environment and customize the graphic user interface
- Be aware of graphic user interface standards

CIP 17 Introduction to Multimedia Lecture (2:2)
Laboratory (1:2)

Prerequisite:
CIP 6

Goal:
Students will learn to develop multimedia instructional software for microcomputers. This will include computer tutorials, interactive video disk applications, and user manuals.

Objectives:
Upon successful completion of this course, the student will be able to:

- Understand the basics of branching computer tutorials

- Be able to select appropriate media, such a video disks for instructional purposes
- Develop simulations and models of reality for instruction
- Consider the benefits and constraints of non-human delivery systems.