

SIMON FRASER UNIVERSITY

MEMORANDUM

S.84-71

To..... SENATE.....
.....
Subject..... CURRICULUM CHANGES - COMPUTING
SCIENCE.....

From..... SENATE COMMITTEE ON UNDERGRADUATE
..... STUDIES.....
Date..... NOVEMBER 14, 1984.....

Action undertaken by the Senate Committee on Undergraduate Studies at its meeting of November 13, 1984 gives rise to the following motions:

MOTION 1:

"That Senate approve and recommend approval to the Board of Governors, as set forth in S.84-71 , the proposed new courses

CMPT 275 Software Engineering
CMPT 484 Compiler Construction"

MOTION 2:

"That Senate approve and recommend approval to the Board of Governors, as set forth in S.84-71 , the proposed deletion of courses

CMPT 118	Computing Projects in the Arts and Sciences
CMPT 121	Computing Project - Mathematics
CMPT 131	" " - Chemistry
CMPT 132	" " - Physics
CMPT 141	" " - Biology
CMPT 142	" " - Kinesiology
CMPT 151	" " - Geography
CMPT 152	" " - Archaeology
CMPT 161	" " - Anthropology
CMPT 162	" " - Communication
CMPT 163	" " - History
CMPT 164	" " - Political Science
CMPT 165	" " - Sociology
CMPT 171	" " - Business Administration
CMPT 172	" " - Economics
CMPT 181	" " - Languages
CMPT 182	" " - Linguistics
CMPT 183	" " - English
CMPT 184	" " - Literature
CMPT 185	" " - Design
CMPT 186	" " - Music
CMPT 492	Microprogramming and Emulation"

MOTION 3:

"That Senate approve and recommend approval to the Board of Governors, as set forth in S.84-71, the program changes listed in the submission from the Department."

FOR INFORMATION:

Acting under delegated authority at its meeting of November 13, 1984 the Senate Committee on Undergraduate Studies approved, for the following courses, the changes as set out in the submission from the Department:

CMPT 101
CMPT 104
CMPT 105
CMPT 201
CMPT 218
CMPT 260
CMPT 290
CMPT 291
CMPT 350
CMPT 354
CMPT 371
CMPT 383
CMPT 390
CMPT 391
CMPT 400
CMPT 401
CMPT 483
CMPT 491
CMPT 495

SIMON FRASER UNIVERSITY

MEMORANDUM

SCUS 84-30

To..... Mr. R. Heath, Registrar &
Secretary to the Senate
..... Committee on Undergraduate
Studies
Subject..... CURRICULUM CHANGES, COMPUTING
SCIENCE (ISC 84-23)

From..... Janet Blanchet, Secretary.....
Faculty of Interdisciplinary
..... Studies Undergraduate Curriculum
Committee
Date..... November 1, 1984.....

At a meeting of the Faculty of Interdisciplinary Studies Undergraduate Curriculum Committee held on Tuesday, October 30 members of the committee approved the curriculum changes outlined in the attached package. Would you please put this item on the agenda of the November 13, 1984 meeting of the Senate Committee on Undergraduate Studies.

JB/rj
Encl:

Janet Blanchet

To: Ted Dobb, University Librarian
From: Rob Cameron, Director of the Undergraduate Program,
Computing Science
Date: October 30, 1984

Enclosed are course proposal forms for CMPT 275, CMPT 483 and CMPT 484, for information. CMPT 275 is a new course which is proposed to replace CMPT 118. CMPT 483 is a revision of an existing course, with part of its material being split off to create a new course CMPT 484. The proposed courses have been approved by the Faculty of Interdisciplinary Studies Undergraduate Curriculum Committee.

To: Faculty Curriculum Committee Chairpersons
From: Rob Cameron, Director of the Undergraduate Program,
Computing Science
Date: October 30, 1984

Enclosed are course proposal forms for CMPT 275, CMPT 483 and CMPT 484, for information. CMPT 275 is a new course which is proposed to replace CMPT 118. CMPT 483 is a revision of an existing course, with part of its material being split off to create a new course CMPT 484. The proposed courses have been approved by the Faculty of Interdisciplinary Studies Undergraduate Curriculum Committee.

(ISC 84-23)

To: Janet Blanchet, Interdisciplinary Studies
From: Rob Cameron, Director of the Undergraduate Program,
Computing Science
Date: October 30, 1984

Attached is the revised copy of the memo regarding Computing Science curriculum proposals (CMPT-UGCC.84-3:14, ISC 84-23), as you requested. The revisions are as presented in the FISC meeting of today, namely that item 19 in the "Summary of Changes" should include a change in title for CMPT 483, and that the "Rationale" for detailed curriculum revision proposal #3 (CMPT 205 Required for Computing Minors) should indicate that CMPT 260 is no longer a part of the lower level requirements for Computing Minors.

Also enclosed are the course proposal forms for CMPT 275, 483 and 484 with references added and as signed by our Departmental Chairman.

Oct. 22, 1984

Simon Fraser University

To: F.I.D.S. Undergraduate Curriculum Committee Members
From: Rob Cameron, Director of the Undergraduate Program,
Computing Science Department
Date: October 22, 1984

The following are proposed curriculum revisions for Computing Science as amended and approved by the Computing Science Undergraduate Curriculum Committee. Formal departmental approval of the proposals is expected by October 30, 1984.

Summary of Changes

Course Changes

1. Courses to be deleted: CMPT 118, 121, 131, 132, 141, 142, 151, 152, 162, 163, 164, 165, 171, 172, 181, 182, 183, 184, 185, 186, 492, 161.
2. New courses: CMPT 275, CMPT 484.
3. CMPT 101: change in prerequisites and calendar description.
4. CMPT 104: change in calendar description.
5. CMPT 105: change in prerequisites.
6. CMPT 201: change in prerequisites.
7. CMPT 218: change in prerequisites.
8. CMPT 260 renumbered as CMPT 320.
9. CMPT 290: change in calendar description.
10. CMPT 291: change in credits, prerequisites and calendar description.
11. CMPT 350: change in prerequisites.
12. CMPT 354: change in prerequisites.
13. CMPT 371: change in prerequisites.
14. CMPT 383: change in prerequisites.
15. CMPT 390: change in prerequisites and calendar description.
16. CMPT 391: change in prerequisites and calendar description.
17. CMPT 400: change in prerequisites and calendar description.
18. CMPT 401: change in prerequisites.
19. CMPT 483: change in credits, prerequisites, calendar description and title.
20. CMPT 491: change in prerequisites and calendar description.
21. CMPT 495: change in prerequisites.

Program Changes

1. CMPT 275 replaces CMPT 118 as a lower division requirement in the Computing Science Major and Honors Programs and in the Digital Systems Design Honors Program.
2. CMPT 260 is removed as a lower division requirement for the Computing Science Major and Honors Programs and the Digital System Design Honors Program.
3. An upper division "Social Aspects of Computing" requirement is added for the Computing Science Major and Honors Programs and the Digital System Design Honors Program.
4. CMPT 390 replaces CMPT 400 as the Key Course in the area of

- Computer Design and Organization for the Computing Science Major and Honors Programs.
5. CMPT 350 is removed as a course in the area of Information Systems for the Computing Science Major and Honors Programs.
 6. CMPT 484 is introduced as a course in the area of Software Systems and Programming for the Computing Science Major and Honors Programs.
 7. CMPT 118 is replaced as a requirement in the Computing Science Minor Program by a requirement for either CMPT 275 or CMPT 290.
 8. CMPT 205 becomes a required course in the Computing Science Minors Program.
 9. A breadth requirement is added to the Computing Science Minor Program.

Detailed Curriculum Revision Proposals

Here follows the list of detailed curriculum revision proposals as approved by the Computing Science Undergraduate Curriculum Committee.

1. Lower Division Changes

The following changes to lower division courses are proposed.

- a. The prerequisite to CMPT 101 (Introduction to a High Level Programming Language A) is changed to include B.C. High School Computer Science 11 (or 12) with a grade of B or higher, or CMPT 001 (Computers and the Activity of People) or equivalent computing experience (with permission of the department).

- b. The calendar description of CMPT 101 is changed as follows.

An intensive introduction to high level programming in a modern programming language. Review of fundamental programming concepts, including integer and real numbers as data objects, variables, assignment, conditional statements and loops. The concept of an algorithm. Structured programming using subprograms, modules and recursion. Structured data objects including arrays, strings and records. Program and user documentation.

- c. CMPT 101 or CMPT 104 (Introduction to a High Level Programming Language II) become corequisite rather than prerequisite to CMPT 105 (Fundamental Concepts of Computing). CMPT 103 (Introduction to a High Level Programming Language B) with a grade of B or higher is no longer sufficient as a prerequisite to CMPT 105.

- d. The calendar description for CMPT 104 is modified by deleting the phrase "and CMPT 103".
- e. CMPT 275 (Software Engineering) is introduced as a new course. See the attached new course proposal (CMPT-UGCC.84-3:10).
- f. CMPT 275 replaces CMPT 118 (Computing Projects in the Arts and Sciences) as a lower division requirement in the Computing Science Major and Honors Programs and the Digital Systems Design Honors Program.
- g. CMPT 118 as a lower division requirement in the Computing Science Minor Program is replaced by a requirement for either CMPT 275 or CMPT 290 (Introduction to Digital Systems).
- h. CMPT 118 is removed as a prerequisite to CMPT 201 (Program and Data Organization).
- i. CMPT 201 replaces CMPT 118 as a prerequisite to CMPT 218 (Special Topics in Computing Science).
- j. CMPT 118 is deleted.

Rationale

These changes rationalize and streamline the lower division requirements in Computing Science. Many students are now entering Computing Science with significant exposure to the field, either through B.C. High School courses or other avenues. Currently, students with such a background find some material in CMPT 101 boring while students without this background often have difficulty keeping pace. The new prerequisites to CMPT 101 will allow the course to be taught at a level challenging to those with previous exposure to the area. CMPT 001 will be used as a first course in Computing for the diminishing numbers of intended Computing Majors with no previous exposure to Computing.

The new prerequisites for CMPT 101 and the consequent strengthening of CMPT 101 allow it to be used as a corequisite rather than a prerequisite to CMPT 105.

The strengthening of CMPT 101 also allows the elimination of CMPT 118 which has become a problem area in our lower division curriculum. Originally intended as a project course to supplement a two-credit introductory course in computing, 118 has evolved into a course with both a project and a lecture component. Unfortunately, with the strengthening of the introductory course from two to four credits, the projects undertaken in CMPT 118 have become so large that they require material from CMPT 201. This material has been provided on an as-needed basis in the lecture component of 118, resulting in significant, but unpredictable (i.e., depending on the

instructor's choice of project) overlap with 201. At the same time, the lecture component of 118 has been used to introduce students to the basic concepts of software engineering, but this has been unsuccessful because the students do not seem ready to deal with these concepts after only one course in computing. This latter problem is solved by placing this material in CMPT 275, taken after students have had exposure to the fundamental material of CMPT 101, 105, 201 and 205. Students will thus be able to proceed directly into CMPT 201 (Data and Program Organization) after completing CMPT 101.

These changes also meet the spirit of the PACUP recommendations for streamlining the lower division requirements in Computing Science. Currently, Computing students can take CMPT 101 in their first semester, CMPT 105 and 118 in their second semester and then generally need two more semesters to complete CMPT 201, 205, 260 and 290. With the proposed changes, however (including changes to the Social Aspects of Computing requirement), students entering with sufficient prerequisites from high school could take CMPT 101 and 105 in their first semester, CMPT 201 and 205 in their second semester, and CMPT 275 and 290 in their third semester, completing their lower division computing requirements.

Replacing the Computing Minor requirement of CMPT 118 with a requirement for either CMPT 275 or CMPT 290 gives such students different options for upper division courses depending on prerequisites.

Implementation

All changes are effective for the Fall term 1985. However, students completing CMPT 118 prior to the Fall term 1985 will be exempted from the CMPT 275 requirement.

2. Upper Division Social Aspects Of Computing Requirement

The following changes are proposed to move the social aspects of computing requirement from the lower to the upper division.

- a. CMPT 260 (Social Implications of a Computerized Society) is removed as a prerequisite to CMPT 350 (Information and Public Policy).
- b. CMPT 260 is renumbered as CMPT 320.
- c. For the Computing Science Major and Honors Programs and the Digital System Design Honors Program, the lower division requirement of CMPT 260 is eliminated and in its place an upper division "Social Aspects of Computing" requirement is substituted. The "Social Aspects of Computing" requirement is that students must take one of the following courses: CMPT 320 or CMPT 350.

- d. For the Computing Science Major, Minor and Honors Programs, CMPT 350 is removed as a course that can be used for requirements in the Information Systems area.

Rationale

CMPT 260 has been used to fulfill the general need to expose students to the interaction between computer technology and society at large. Appreciation of this material requires considerable maturity on the part of the student; however; moving this requirement to the upper division is thus appropriate.

At the same time, there is more than one way to explore the relationship between computer technology and the public, so it makes sense to allow students various alternatives in meeting this requirement. The changes to CMPT 350 allow it to be used as such an alternative.

These changes also meet a major PACUP recommendation for streamlining the lower division requirements in Computing Science, i.e., reducing the number of CMPT courses students must take before the question of their admittance to the program is decided.

Implementation

These changes become effective for the Fall term 1985. Students completing CMPT 260 prior to Fall 1985 will have their upper division "Social Aspects of Computing Requirement" waived.

3. CMPT 205 Required for Computing Minors

It is proposed that CMPT 205 become a required course for Computing Minors.

Rationale

Currently, Computing Minors must take one of CMPT 205 (Introduction to Formal Topics in Computing Science) and CMPT 260 (Social Implications of a Computerized Society). As the Computing Science discipline matures, however, the fundamental topics covered in 205 are becoming increasingly important to the upper division courses in Computing Science. In fact, there remains only a small number of upper division Computing courses for which CMPT 205 is not prerequisite. It is therefore proposed that CMPT 205 becomes a requirement for the Computing Science Minor program. CMPT 260 would no longer be a lower level requirement for Computing Minors.

Implementation

The new requirement will be waived for students who have declared a Computing Minor prior to the Fall term of 1985.

4. CMPT 205 Prerequisite for CMPT 354

It is proposed that, subject to CMPT 205 becoming a required course for Computing Minors, that the prerequisite to CMPT 354 become CMPT 205 or equivalent.

Rationale

Recently, CMPT 205 was included as a recommended course for CMPT 354 (File and Database Structures), but was not included as a prerequisite because that would unduly restrict the number of upper level Computing courses available to minors who did not choose to take CMPT 205. Since CMPT 205 is now being proposed as a required course for Computing Minors, this objection should soon be overcome.

Implementation

This requirement becomes effective in the Fall term of 1985. For students in the Management and Systems Science program, MATH 243 will be taken as the equivalent of CMPT 205 for the purposes of entry into CMPT 354. This is because MATH 243 may substitute for CMPT 205 in the lower division requirements of that program and CMPT 354 is a required upper division course in that program.

5. CMPT 205 Prerequisite for CMPT 383

It is proposed that, subject to CMPT 205 becoming a required course for Computing Minors, CMPT 205 become a prerequisite for CMPT 383.

Rationale

The study of programming languages is seeing the increased use of formal techniques to complement the traditionally informal methods for analyzing and comparing programming languages. Proper inclusion of this material in CMPT 383 (Comparative Programming Languages) requires that students have a solid introduction to formal topics in Computing Science.

Implementation

This requirement becomes effective in the Fall term of 1985.

6. Rationalization of CMPT 290 and CMPT 291

The following changes are proposed for CMPT 290 and CMPT 291.

- a. The calendar description of CMPT 290 is changed as follows.

Digital circuit design principles for small, medium and large scale integrated circuit building blocks. Switching theory, finite-state machines, introductory register-transfer level design. A sequence of lab experiments parallel and augment the lecture material.

- b. CMPT 291 becomes a 1 credit-hour course.

- c. CMPT 290 becomes a co-requisite to CMPT 291. Explicit mention of CMPT 105 as a prerequisite to CMPT 291 is no longer necessary.

- d. The calendar description of CMPT 291 is changed as follows.

This course augments CMPT 290, providing additional material emphasizing the physical principles underlying digital circuits and their influence on the organization and performance limits of digital systems.

Rationale

Currently, CMPT 290-3 (Introduction to Digital Systems) and CMPT 291-4 (Introduction to Digital Circuit Design) cover more or less the same material, except that CMPT 291 deals with the electronics underlying digital circuitry in much more detail than does CMPT 290. The proposed change eliminates the duplication of material in these two courses. CMPT 290 is modified to better prepare students for upper level hardware courses by strengthening its treatment of gate level and register-transfer level digital circuitry while reducing its emphasis on the underlying electronics. CMPT 291 becomes a 1 credit-hour course which provides the electronics underlying digital circuitry and which is normally taken in conjunction with CMPT 290.

These changes also meet the spirit of the PACUP recommendations for streamlining the lower division requirements in Computing Science.

Implementation

This change has no effect on other courses and will be implemented for the Fall term of 1985.

7. Changes to CMPT 390 and CMPT 400

The following changes are proposed for CMPT 390 and CMPT 400.

- a. CMPT 390 becomes prerequisite to CMPT 400. The current CMPT 400 prerequisite of CMPT 290 or CMPT 291 is no longer explicitly listed.
- b. CMPT 390 replaces CMPT 400 as a prerequisite to CMPT 401.
- c. CMPT 390 replaces CMPT 400 as the Key Course in the area of Computer Design and Organization.
- d. CMPT 290 replaces CMPT 291 as a prerequisite to CMPT 390.
- e. The calendar description of CMPT 390 is changed to read as follows.

Review of introductory register-transfer level design. Construction of the basic subsystems of computers (control unit, data paths, memory, input/output). Assembly of subsystems into basic computer architectures.

- f. The calendar description of CMPT 400 is changed to read as follows.

Alternative computer architectures are explored. Topics include evolution of computer architectures, stack, SIMD and MIMD architectures, micro-architecture, performance enhancement techniques and current topics in computer architecture.

- g. CMPT 390 replaces CMPT 400 as a required course in the Mathematics and Computing Science Joint Honors Program and CMPT 400 is listed in place of CMPT 390 as one of the options for the additional required course in the Computer Design and Organization area of the Mathematics and Computing Science Joint Honors Program.

Rationale

Currently, students entering CMPT 400 (Hardware Architecture) do not have sufficient background to allow the full exploration of alternative computer architectures. It is therefore proposed that the prerequisite to CMPT 400 be strengthened to include CMPT 390 (Digital Circuits and Systems), which in turn is to be modified to include the basics of conventional computer architectures. This modification of CMPT 390 also serves to make it a sufficient introduction to computer architecture for the purposes of CMPT 401 (Operating Systems).

In order that these changes do not impose additional upper level requirements on Computing Majors, the Key Course in the area of Computer Design and Organization is proposed to be changed to CMPT 390 from CMPT 400. Given the modification of CMPT 390 to include the basics of conventional computer architecture, 390 does take on central importance to the area. This change in turn requires that CMPT 390 be accessible to CMPT Majors who have taken CMPT 290 but not CMPT 291; this is appropriate, however, since the material of CMPT 390 no longer requires knowledge of the underlying electronics of digital hardware as emphasized in CMPT 291.

The calendar description of CMPT 400 is modified to remove some topics to be covered in CMPT 390 and to add some new topics.

The introduction of CMPT 390 in place of CMPT 400 as the required hardware course for the Mathematics and Computing Science Joint Honors Program is a logical extension to the changes above. This proposal will be forwarded to the Mathematics and Computing Science committee; Ron Harrop has already been contacted and has indicated that the change will very likely be approved.

Implementation

All changes become effective for the Fall term 1985. Students having taken CMPT 400 before Fall 1985 will be exempted from taking CMPT 390 both as a prerequisite to CMPT 401 and as the Key Course in the area of Computer Design and Organization.

8. Changes to CMPT 391, CMPT 491 and CMPT 495

The following changes to upper division hardware courses are proposed.

- a. CMPT 390 becomes a co-requisite to CMPT 391.
- b. CMPT 391 becomes a Laboratory/Lecture course.
- c. CMPT 391 replaces CMPT 390 as a prerequisite to CMPT 491.

- d. The calendar description for CMPT 491 is changed as follows.

Advanced topics in analogue and digital circuit design. Topics include operational amplifiers, phase-lock loops, interface design, bus design and protocols, transmission lines and reflections, new logic technologies, and other current topics in digital circuit design.

- e. The prerequisites to CMPT 495 are changed to be CMPT 391 with CMPT 491 recommended.

Rationale

CMPT 391 (Microcomputer Hardware Workshop) is a course in applying the digital circuit theory developed in CMPT 390 (Digital Circuits and Systems), thus making CMPT 390 a logical co-requisite to CMPT 391. CMPT 391 is also changed to include a lecture component, however, to provide material related to the practical details of implementing working digital circuitry.

The prerequisites for CMPT 491 (Analogue and Digital Circuits) are strengthened to provide a greater background in digital hardware and also to give students practical experience to appreciate some of the advanced analogue and digital hardware design concerns. The calendar description is updated to avoid conflict with CMPT 400.

CMPT 391 becomes the new prerequisite to CMPT 495 (Digital System Design and Specification Laboratory) to ensure that students have not only the basic theoretical background of CMPT 390, but also have the basic practical experience provided by 391.

Implementation

All changes become effective for Fall 1985.

9. CMPT 492 deleted

It is proposed that CMPT 492 (Microprogramming and Emulation) be eliminated.

Rationale

This course is no longer needed as the topics it covers are dealt with elsewhere (in CMPT 390, CMPT 400 and other courses). This also meets PACUP recommendations for program rationalization.

10. Deletion of Short Project Courses

Moved Rob Cameron, seconded Hassan Reghbati that the following one-credit project courses be deleted: CMPT 121, 131, 132, 141, 142, 151, 152, 162, 163, 164, 165, 171, 172, 181, 182, 183, 184, 185, 186.

Rationale

There is little or no demand for these courses. Deletion of these courses meets the spirit of PACUP recommendations for Computing Science.

11. Math 152 Prerequisite to CMPT 371

It is proposed that Math 152 (Calculus II) be made prerequisite to CMPT 371 (Data Communications and Networking).

Rationale

The analysis of data networks requires some knowledge of calculus. Students without this background are having difficulty in the course.

Implementation

This change is effective for the Fall term 1985.

12. Splitting of CMPT 483

It is proposed that CMPT 483 (Compiler Design) be split into two courses as follows.

- a. CMPT 483 is changed according to the attached revised course proposal (CMPT-UGCC.84-3:11).
- b. CMPT 484 is introduced as an advanced compiler construction course per the attached new course proposal (CMPT-UGCC.84-3:12).
- c. For the Computing Science Major, Minor and Honors Programs and the Mathematics and Computing Science Joint Honors Program, CMPT 484 becomes a new option for meeting requirements in the Software Systems and Programming area.

Rationale

Currently, CMPT 483 contains too much material for a one semester course. In particular, it is difficult for students to complete course projects in one term. Splitting of CMPT 483 into two courses should alleviate this problem, and should also allow the incorporation of additional material not currently covered in CMPT 483.

Implementation

These changes are effective for the Fall term 1985.

13. Change to Upper Level Requirements for Computing Minors

It is proposed that the upper division course requirements for Computing Minors be changed as follows (changes underlined).

Students minoring in Computing Science must complete at least 15 credits of upper division Computing Science (CMPT or MACM) courses shown in Table I, of which no more than 9 credits can be from any one area. Normally at least 12 of these credits will be in CMPT courses.

Rationale

This prevents Computing Minors from having too narrow a focus by taking courses in only one area of Table I (1984-5 calendar, pp. 113-114). It also ensures that only technical electives as listed in Table I count towards a Computing Minor.

Implementation

This change is effective for all students declaring a CMPT Minor in the Fall term of 1985 or later. This change also applies to declared minors who have completed less than 9 credits of the required upper division CMPT courses by Fall 1985.

SENATE COMMITTEE ON UNDERGRADUATE STUDIES

NEW COURSE PROPOSAL FORM

Department: COMPUTING SCIENCE

1. Calendar Information

Abbreviation Code: CMPT Course Number: 275

Credit Hours: 3 Vector: 3-0-2

Title of Course: SOFTWARE ENGINEERING

Calendar Description of Course:

The software life cycle: requirements/ specification, design, implementation, checkout, maintenance. Software tools: requirement specification languages, program design languages, program editors, program transformation systems, test data generators, automatic verifiers. Integrated programming environments.

Nature of Course

Prerequisites (or special instructions):

CMPT 201, 205

What course (courses), if any, is being dropped from the calendar if this course is approved:

CMPT 118

2. Scheduling

How frequently will the course be offered? Every semester

Semester in which the course will first be offered? 86-1

Which of your present faculty would be available to make the proposed offering possible? Several

3. Objectives of the Course

To introduce students to the problems and techniques involved in the design of economical and reliable large-scale software systems.

4. Budgetary and Space Requirements (for information only)

What additional resources will be required in the following areas:

Faculty

Staff

Library

NONE

Audio Visual

Space

Equipment

5. Approval

Date: Oct. 30, 1987

[Signature]
Department Chairman

[Signature]
1 Nov 87
Dean

[Signature]
Chairman, SCUS

SCUS 73-34b: (When completing this form, for instructions see Memorandum SCUS 73-34a.)

CMPT 275 - Software Engineering

Sample Course Outline

The Software Life Cycle

Requirements/specification, design, implementation,
checkout, maintenance

Costs of software development, maintenance and use

The importance of reliability

The importance of readability vs. writability

The two-stage approach to development of efficient software

Software tools

Documentation

Requirements/Specification

The importance of clear, complete specifications

Requirements specification languages

Design

Top-down design

Data-driven design

Design Methodologies

Program Design Languages (PDLs)

Implementation

Top-down implementation

Bottom-up implementation

Program editors

Program transformation systems

Checkout

Desk checking and code reviews

Test data generators

Automatic verification

Maintenance

Integrated Programming Environments

Reference: Glass, Robert L., Software Reliability Guidebook,
Prentice-Hall, 1979.

SENATE COMMITTEE ON UNDERGRADUATE STUDIES

REVISED ~~NEW~~ COURSE PROPOSAL FORM

1. Calendar Information

Department: COMPUTING SCIENCE

Abbreviation Code: CMPT Course Number: 483

Credit Hours: 3 Vector: 3-0-0

Title of Course: PARSING AND INTERPRETATION

Calendar Description of Course:

Theoretical and practical aspects of parsing and interpreter implementation. Analysis of the structure of high level languages, lexical analysis, parsing, syntax error recovery, internal representation and environment for interpretation, interactive system requirements.

Nature of Course

Prerequisites (or special instructions):

CMPT 205, 275, 383, MATH 306

What course (courses), if any, is being dropped from the calendar if this course is approved:

Replaces existing version of CMPT 483. Reduction in credit and content.

2. Scheduling

How frequently will the course be offered? At least once per year.

Semester in which the course will first be offered? 85-3

Which of your present faculty would be available to make the proposed offering possible? J.J. Weinkam, R. Cameron, H. Reghbati

3. Objectives of the Course

To introduce basic theory of regular and context free languages, parsing, and the practical aspects of the implementation of parsers and interpreters. Students will develop a working interpreter as part of the course.

4. Budgetary and Space Requirements (for information only)

What additional resources will be required in the following areas:

Faculty NONE

Staff TEACHING ASSISTANT (1/2)

Library NONE

Audio Visual NONE

Space NONE

Equipment NONE. COMPUTING RESOURCES TYPICAL FOR PROJECT ORIENTED COMPUTING COURSES ON MTS OR OS SYSTEM.

5. Approval

Date: Oct. 30, 1984

1 Nov 84

il Carone
Department Chairman

J. D. Chabot
Dean

Chairman, SCUS

CMPT 483 - Parsing and Interpretation

Sample Course Outline

Overview of Languages, Interpreters and Compilers

- Translation of languages
- Comparison of interpreters and compilers
- Lexical analysis, parsing, intermediate representation, interpretation, error recovery

Programming Languages

- Structure of languages
- Lexical structure
- Syntactic structure
- Execution environment

Lexical Analysis

- Finite automata and regular expressions
- Lexical analysis for practical languages

Syntax Analysis

- Context free grammars
- Derivation and parse trees
- Bottom-up parsing
- Top-down parsing
- Syntax error recovery

Execution

- Immediate execution
- Internal representation
- Execution environments
- Expression evaluation
- Representation of data objects

Interactive Environments

- Attention handling
- Interactive error recovery
- Dynamic i/o reconfiguration

Text: Aho, Alfred V., and Ullman, Jeffrey D., Principles of Compiler Design, Addison-Wesley, 1978.

SENATE COMMITTEE ON UNDERGRADUATE STUDIES

NEW COURSE PROPOSAL FORM

1. Calendar Information

Department: COMPUTING SCIENCE

Abbreviation Code: CMPT Course Number: 484

Credit Hours: 3 Vector: 3-0-0

Title of Course: COMPILER CONSTRUCTION

Calendar Description of Course:

Theoretical and practical aspects of language translation and compiler implementation, building on the material covered in CMPT 483 Parsing and Interpretation. Translation, intermediate representations, code generation, optimization, run time environments, semantic and execution error handling. Students will design and implement a working compiler for a simple language as a course project.

Nature of Course

Prerequisites (or special instructions):

CMPT 483

What course (courses), if any, is being dropped from the calendar if this course is approved:

2. Scheduling

How frequently will the course be offered? At least once every two years.

Semester in which the course will first be offered? 86-1

Which of your present faculty would be available to make the proposed offering possible? J.J. Weinkam, R. Cameron, H. Reghbat

3. Objectives of the Course

To introduce the basic theory of language translation and compiler design, as well as some of the practical aspects of compiler implementation. Students will develop a working compiler for a simple block-structured language as part of the course.

4. Budgetary and Space Requirements (for information only)

What additional resources will be required in the following areas:

Faculty NONE

Staff TEACHING ASSISTANT (1/2)

Library NONE

Audio Visual NONE

Space NONE

Equipment NONE. COMPUTING RESOURCES TYPICAL FOR PROJECT ORIENTED COMPUTING COURSES ON MTS OR OS SYSTEM.

5. Approval

Date: Oct. 30, 1984

1 Nov 84

Wil Carcone

Robert

Department Chairman

Dean

Chairman, SCUS

SCUS 73-34b: (When completing this form, for instructions see Memorandum SCUS 73-34a.

CMPT 484 - Compiler Construction

Sample Course Outline

Overview of Compilers

- Translation of languages
- Structure of compilers

Review of Lexical Analysis and Parsing

- Lexical analysis
- Bottom-up parsing
- Top-down parsing

Syntax Directed Translation

- Expressions
- Assignments
- Booleans
- Declarations
- Arrays
- Procedure calls, functions
- Scoping
- Control structure

Run-Time Environments

- Storage management
- I/O support
- Debugging facilities
- Library support
- Error handling

Optimization and Code Generation

- Register allocation
- Data flow analysis
- Object program format
- External linkage

Text: Aho, Alfred V., and Ullman, Jeffrey D., Principles of Compiler Design, Addison-Wesley, 1978.

Reference: Aho, Alfred V., and Ullman, Jeffrey D., The Theory of Parsing, Translation and Compiling, Vols I and II, Prentice-Hall, 1972.