S.74-49

SIMON FRASER UNIVERSITY

MEMORANDUM

To: SENATE

From: SENATE COMMITTEE ON UNDERGRADUATE STUDIES

Subject: NEW COURSE PROPOSALS:
CMPT 103-3 - INTRODUCTION TO A HIGH LEVEL PROGRAMMING LANGUAGE I
CMPT 104-3 - INTRODUCTION TO A HIGH LEVEL PROGRAMMING LANGUAGE II

Date: APRIL 18, 1974

MOTION 1: "That Senate approve, as set forth in S.74-49, the new course proposals for:

CMPT 103-3 - Introduction to a High Level Programming Language I
CMPT 104-4 - Introduction to a High Level Programming Language II."

MOTION 2: "That Senate waive the normal two semester time lag requirement in order that these two course may be first offered in the Fall semester 74-3."

S.74-49

SIMON FRASER UNIVERSITY

MEMORANDUM

To SENATE	From Senate Committee on Undergraduate Studies
Subject Faculty of Interdisciplinary Studies-New Course Proposals	Date April 19, 1974

The Senate Committee on Undergraduate Studies has examined the attached course proposals for Computing Science 103-3:Introduction to High Level Programming Language I; Computing Science 104-1: Introduction to a High Level Programming Language II and recommends their approval to Senate. The Committee further recommends that, if Senate approves the courses in question, the normal two semester time lag be waived to enable these courses to be offered during the Fall 1974.

No question arose in the Committee concerning the academic acceptability of these courses; but there was some question about their overlap with courses currently offered by the Mathematics Department, in particular Mathematics 106. The Director of the Computing Science Program and the Dean of Interdisciplinary Studies were requested to discuss this question with the Mathematics Department and, on the understanding that that Department had no objection to these new courses and envisaged the withdrawal of Mathematics 106 from their program, the courses were approved. The attached memorandum from the Chairman of the Department of Mathematics addresses itself to this question.

I. Mugridge

:ams

att.

SIMON FRASER UNIVERSITY

MEMORANDUM

To Dr. I. Mugridge
Assistant Vice-President
Vice-President, Academic

From Dr. A.H. Lachlan
Chairman
Mathematics Department

Subject New courses in Computer Science:
Overlap with Mathematics 106

Date April 19, 1974

This is in reply to your query regarding the new courses CMPT.103 and CMPT. 104 proposed by the Computing Science Programme. Insofar as FORTRAN is treated in these courses, there certainly is overlap with Mathematics 106. However it is not forseen by this Department that Mathematics 106 will ever be offered again. Under the two year rule if it is not offered Mathematics 106 will eventually be struck from the calendar and in any case it will not be offered again as part of our regular course offerings. Thus, I feel that the fact that there is overlap with Mathematics 106 should not constitute any kind of obstacle to the approval of these two new courses.

A.H. Lachlan

A.H. Lachlan

AHL/lh

cc: Dr. Aronoff
Dean of Science
Dr. J. Weinkam
Computer Science

SIMON FRASER UNIVERSITY SCUS 74-14

MEMORANDUM

To Mr. H. Evans, Registrar and
Secretary of the Senate Committee
onUndergraduate Studies.

From J. Blanchet, Secretary,
Faculty of Interdisciplinary Studies
Curriculum Committee.

Subject CMPT 103-3 & CMPT 104-1.

Date March 15/74.

The attached new course proposals have been approved by the Faculty of Interdisciplinary Studies Curriculum Committee, and are now forwarded to you for consideration by the Senate Committee on Undergraduate Studies. Would you please place them on the agenda.

Attachment.

J. Blanchet

I.S.C. 74-3.

SENATE COMMITTEE ON UNDERGRADUATE STUDIES

NEW COURSE PROPOSAL FORM

1. Calendar Information

Department: *Computing Science Program*

Abbreviation Code: *CMPT* Course Number: *103* Credit Hours: *3* Vector: *3-0-0*

Title of Course: *Introduction to a High Level Programming Language I*

Calendar Description of Course: *This course introduces the student to a high level programming language. Major Programming Languages are offered including PL/1, COBOL, and FORTRAN. PL/1 serves as the key language for most subsequent courses in the Computing Science Program; COBOL is an important language for business applications; and FORTRAN is still widely used in scientific applications. The student should consult the program advisor for a list of currently available languages and for advice in selecting the language most suited to his program.*

Nature of Course

Guided self-study

Prerequisites (or special instructions):

no prerequisites

What course (courses), if any, is being dropped from the calendar if this course is approved: *CMPT 102*

2. Scheduling

How frequently will the course be offered? *Every semester*

Semester in which the course will first be offered? *Fall 1974*

Which of your present faculty would be available to make the proposed offering possible? *Ms. Doreen Godwin*

3. Objectives of the Course *See attached.*

4. Budgetary and Space Requirements (for information only)

What additional resources will be required in the following areas:

Faculty *None*

Staff *None*

Library *None*

Audio Visual *None*

Space *None*

Equipment *None*

5. Approval

Date: March 13/74. March 18, 74

Department Chairman

R C Brown

Dean

Chairman, SCUS

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

SENATE COMMITTEE ON UNDERGRADUATE STUDIES

NEW COURSE PROPOSAL FORM

1. Calendar Information
Department: Computing Science Program

Abbreviation Code: CMPT Course Number: 104 Credit Hours: 1 Vector: 3-0-0

Title of Course: *Introduction to a High Level Programming Language II*

Calendar Description of Course: *This course is identical to CMPT 103-3 and is intended for the student who wishes to learn a second high level language under supervision and for academic credit. It is considerably easier to master a second high level language; therefore this course carries only one credit.*

Nature of Course *Guided self-study*

Prerequisites (or special instructions): *The student must select a different language from the one that he studied in CMPT 103.*

What course (courses), if any, is being dropped from the calendar if this course is approved: *CMPT 102*

2. Scheduling

How frequently will the course be offered? *Every semester*

Semester in which the course will first be offered? *Fall 1974*

Which of your present faculty would be available to make the proposed offering possible? *Ms. Doreen Godwin*

Objectives of the Course *See attached*

4. Budgetary and Space Requirements (for information only)

What additional resources will be required in the following areas:

Faculty *None*

Staff *None*

Library *None*

Audio Visual *None*

Space *None*

Equipment *None*

5. Approval

Date: 3/15/74   March 18, 74   ______

T. Sterling   R C Brown   ______

Department Chairman   Dean   Chairman, SCUS

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

Attachment for CMPT 103 and CMPT 104 Course Proposals.

3. Objectives of the Course

3(a) The purpose of the course is to make the student into a reasonably competent programmer in the selected programming language. By the end of the course the student will be able to take a reasonably simple programming problem and define the necessary input/output requirements, prepare a flowchart of an algorithm to solve the problem, write the program in the selected programming language, debug it, and produce documentation specifying how the program can be used and the overall methods by which it achieves its objectives.

The course content includes the format and function of the major statements in the programming language, and the methods by which common programming techniques such as looping, decision making, construction of subroutines, and input and output are achieved in the language. The student masters this material by working his way through a suitable textbook covering one major topic at a time. For each topic to be covered, the student is given a reading assignment and a self-study quiz to test his understanding of the assigned material. Upon completion of the reading assignment and quiz, the student consults with the instructor who assigns an appropriate programming assignment to allow the student to obtain the necessary experience in using the language features he has studied. In the later part of the course the programming assignments are designed to integrate in a single assignment several of the language features previously studied separately.

3(b) Knowledge of a high level programming language is a necessary prerequisite for any further work in computing science.

3(c) This course replaces CMPT 102. It does not overlap any other existing courses.

Rationale for the proposed courses CMPT 103 and CMPT 104

Teaching an introductory course in a high level programming language presents two problems: One problem is that there are a number of languages that a student may legitimately wish to study as his first exposure to high level language programming. For example students in Economics and Commerce need to learn the COBOL language, whereas science students desiring only to deal with numerical problems using one of the available subroutine packages need to study FORTRAN. On the other hand, because of its greater scope and versatility, PL/1 is the major language used in most advanced courses in the Computing Science Program. The second problem is that regardless of the language chosen, students come to such a course with a wide range of backgrounds and aptitudes. When the course is offered in a traditional lecture format, only one language may be taught at a time; and, regardless of the pace selected, it is too slow for many students and too fast for others.

Interviews of students studying a high level language for the first time reveal that most of the material is mastered by reading the textbook and putting the material presented into practice through programming assignments. Students actually learn relatively little material in the lectures; but they do have a need to ask specific questions to clear up individual misunderstandings.

An instructional technique which allows each student to read the text and solve programming problems at his own pace with the aid of a carefully prepared self-study guide overcomes both of these problems and, at the same time, frees the instructor to deal with individual questions and misunderstandings. This method of teaching introductory high level language programming has been tried experimentally during the Spring 1974 semester in the course CMPT 102. Two sections of the course were scheduled but it was not announced in advance that there would be any difference in instructional technique. One of the sections followed the normal lecture method that has been used in previous offerings of the course. In the other section the guided self-study method was used. (A copy of the self-study guide is attached.)

Both sections used the same text (Pollack, S. and Sterling, T., Essentials of PL/1) and were given the same assignments and tests. The self—study class met with the instructor twice a week at the regularly scheduled time for a work session and to handle any individual problems. This encouraged students to keep up with the recommended time schedule by providing a regularly scheduled period in which the course work could be done. The assignment marks and the relative number of students who has fallen behind schedule were equivalent for the two groups. However in the self-study group three students had completed the course by the mid-semester point and several others were ahead of the suggested schedule. No difference in performance could be observed on the test. The class average of the lecture group was 20.1; that of the self-study group was 20.7.

This method of teaching computer languages is not a new approach but has been used within the computer industry for about five years. It has been found very helpful in accommodating various background knowledge levels and in providing a choice in computer language study.

A similar approach has also been used in teaching other subjects; for example Maltbie, et al (1) describe the use of a "proctorial" system of instruction in Mathematics and give several additional references.

A student evaluation of both sections of CMPT 102 was conducted in the Spring 1974 semester. The departmental assistant and faculty members not involved in teaching the course met with the students and instructor at the regularly scheduled class time for a discussion of the course content, teaching methods, and overall reaction to the course.

The results of the evaluation support our belief that the guided self-study method is the preferable method for teaching high level programming languages. None of the students in either section was opposed to using the self-study method exclusively in the future. On the other hand, virtually all students in both sections strongly felt that the workload of the course was too great for a two credit course. As expected some students in the lecture section complained that the pace was too fast while others had the opposite complaint.

CMPT 100 and CMPT 102 were approved by Senate in 1972 in order to make it possible to offer a minimal introductory sequence to SFU students. While CMPT 102 was intended to teach PL/1 only, CMPT 103, which will replace it, will enable the student to take any of the major programming languages. Initially, study guides are prepared for PL/1, COBOL, or FORTRAN. Students in the mainstream of Computing Science, i.e., either majoring or minoring in the subject, will still be required to take PL/1. Students taking a minimal amount of Computing Science courses mainly to support work in Economics and Commerce will probably prefer to learn COBOL, while such students desiring to work in Science, Mathematics, or those primarily interested in numerical analysis may prefer FORTRAN. Other languages intended for work with text or graphics will be offered as the demand arises.

Because of the large variety of Computer Programming Languages and the different approaches to data representation and problem solving that they engender a student may legitimately wish to learn more than one programming language and should be encouraged to do so. Computing 104-1 is intended for students who wish to learn a second high level language under supervision and for academic credit. This course is identical with CMPT 103 but carries only 1 credit because learning a second high level language is by no means as formidable a task as learning one for the first time. Once the student has mastered two high level languages, however, he will not be able to earn additional academic credit by learning other languages. However, we would encourage students to use the CMPT 103 course materials to study additional languages independently.

Reference

1 Maltbie, A., Savage R., and Wasih J.; The Operation and Evaluation of a Proctorial System of Instruction in Mathematics; American Mathematicsl Monthly, 81: 71-78; 1974.

Self-Study Guide for CMPT 102

Introduction to a High Level Language - PL/1

This course has been set up as a self-study course. This means you may work at your own pace. An instructor will be available to answer questions.

Each chapter of the "Essentials of PL/1" textbook has been divided into a number of steps. A step generally consists of:

1) a reading assignment
2) a self-study quiz
3) a work assignment

The reading assignment introduces you to new material and as such should be done first. Next ask the instructor for the self-study quiz. Do the quiz, correct your answers, and return your work to the instructor. To correct your answers, ask for an "answer sheet". You will also discover that many of the self-study quiz problems have answers in the back of the "Essentials of PL/1" textbook. Any discrepancies between your answers and those on the answer sheet should be reviewed with the advisor. Remember, the self-study quiz is your check to see that the reading assignment has been digested properly; it will not effect your final grade.

Two weeks will be allowed for each assignment. Seventy percent of all assignments must be handed in. Your final grade will be based on assignments plus a final project.

The final project may be one of your own choice or one selected from a list of assignments. Prior approval of your own project by the department will be required.

Remember the instructor is there to assist you. This allows you to concentrate on the areas you do not understand rather than listening to a lecture on areas you do understand. If the study quiz indicates a lack of understanding for that section, further assignments or quizzes may be given.

STUDENT OUTLINE:

CHAPTER I

A\. 1 *Introduction - read pages 1 - 11 (sections 1.1 to 2.2)*

2 *If you are not familiar with Floating Point numbers read the Appendix B, pages 358 - 364.*

3 *See the Instructor for Study Quiz 1*

4 *Read the KEYPUNCHING handout. If after reading the handout you do not feel confident to punch cards, please sign your name on the Keypunching Tutorial Sheet, indicating which tutorial you wish to attend.*

B\. 1 *Read pages 11 - 24 (sections 1.2.3. to 1.2.6.2)*

2 *See the Instructor for Study Quiz 2*

3 *If you are unfamiliar with Flowcharting, read the FLOWCHARTING handout. For further information on Flowcharting, sign your name on the Flowcharting Tutorial Sheet.*

4 *Assignment:*

Draw a flowchart for one of the following operations;

- *i. changing a tire*
- *ii. baking a cake*
- *iii. registering at SFU*

5 *Read the COMPUTER RUN REQUIREMENTS handout. Sign your name on the Computer Run Requirements Tutorial sheet, if you have any doubts about preparing and/or submitting data for the computer run.*

6. *Assignment B:*

(Note: All future assignments will consist of flowcharting, coding and running a problem on the computer. The flowchart and computer output, properly headed with your name, course number and section, must be handed in. (Do not hand in Card Decks)

Flowchart and code either problem 8 or problem 10 (page 36 - 37).

C\. 1 *Read pages 24 - 31 (section 2.6.3 to 1.3)*

2 *See Instructor for Study Quiz 3*

3 *Flowchart and code either problem 15 or problem 16 (pages 38 - 39). (Note: Assume P2/P1 will always be greater than 1 for Problem 15.)*

CHAPTER II

A. 1 Read pages 41 - 50.

2 See Instructor for Study Quiz 4

3 Assignment:
Flowchart, code and run Problem 6 (page 53).

CHAPTER III

A. 1 Read pages 55 - 67 (Sections 111.1 to 111.3.4)
2 See Instructor for Study Quiz 5.
3 Assignment:

Flowchart, code and run one of the following problems 8, 11, 13 or 14 (pages 99 - 101).

B. 1 Read pages 67 - 77 (Sections 111.4.1 to 111.4.5)
Data Aggregates and Arrays
2 See Instructor for Study Quiz 6.
3 Assignment:

Rewrite Problem 13, Chapter 1 (page 37) using arrays for land rates. (Use sample input from 13 for your computer run.)

C. 1 Read Structures pages 77 - 84 (Section 111.4.2)
2 See Instructor for Study Quiz 7.
3 Assignment:

Flowchart and run Problem 11 (page 100).

D. 1 Read Advanced Topics (pages 85 - 96).
2 See Instructor for Study Quiz 8.

CHAPTER IV Internal Manipulations

A. 1 Read pages 103 - 121 (Section IV.1 to IV.2.2)
2 See Instructor for Study Quiz 9.
3 Assignment:
Flowchart and run Problem 18 (page 152)

B. 1 Read pages 121 - 128 (Section IV.3 to IV.3.2.4)
2 See Instructor for Study Quiz 10
3 Assignment:
Flowchart and code Problem 10 (page 150)

C. 1 Read pages 128 - 139 (Sections IV.4 to IV.4.1.5)
2 See Instructor for Study Quiz 11
3 Assignment:
Flowchart and code Problem 24. (Modify to 20 questions to simplify input requirements.)

D. 1 Read Advanced Topics (pages 139 - 146)
2 See Instructor for Study Quiz 12

CHAPTER V Control Operations

A. 1 Read pages 155 - 172 (Sections V.1 to V.2.3)
   2 See Instructor for Study Quiz 13
   3 Assignment:
      Flowchart and code Problem 16 (page 199)

B. 1 Read pages 173 - 188 (Sections V.3 to V.3.3.2)
   2 See Instructor for Study Quiz 14
   3 Assignment:
      Flowchart and code Problem 11 (page 198)

C. 1 Read pages 188 - 194 (Section V.4 to V.4.3)
      Advanced Topics
   2 See Instructor for Study Quiz 15

CHAPTER VI   Stream Oriented Input/Output

A. 1 Read pages 203 - 221 (Sections VI.1 to VI.2.2.4)
    Data Directed Input/Output
  2 See Instructor for Study Quiz 16

B. 1 Read pages 221 - 237 (Sections VI.3 - VI.3.3)
    List Directed Input/Output
  2 See Instructor for Study Quiz 17
  3 Assignment:
    Flowchart and run Problem 8 (page 264) for one array (C)

C. 1 Read pages 237 - 253 (Sections VI.4)
    Edit directed Input/Output
  2 See Instructor for Study Quiz 18
  3 Assignment:
    Flowchart and run <u>one</u> of Problems 14, 19, 21 (pages 267 - 270)
    Note: Sample input cards listed for Problem 20 solution may be used for Problem 19.

D. 1 Read pages 253 - 261 (Sections VI.5 to VI.5.4)
  2 See Instructor for Study Quiz 19

CHAPTER VII   Record Oriented Input/Output

A. 1 Read pages 273 - 283 (Section VII.1 to VII.1.2.3)
2 See Instructor for Study Quiz 20

B. 1 Read pages 284 - 294 (Sections VII.2 to VII.3.3)
2 See Instructor for Study Quiz 21
3 Assignment:
Code only - Problem 4 (page 296)

CHAPTER VIII   Modular Construction of Programs

A. 1 Read pages 297 - 305 (Sections VIII.1 to VIII.1.4)
   2 See Instructor for Study Quiz 22

B. 1 Read pages 305 - 314 (Sections VIII.2 to VIII.2.1.3)
   2 See Instructor for Study Quiz 23

C. 1 Read pages 314 - 321 (Sections VIII.2.2 to VIII.3.2)
   2 See Instructor for Study Quiz 24
   3 Assignment:
      Flowchart, Code and Run <u>one</u> of the following Problems:
   7, 12, 14, 15 or 17

CHAPTER IX

A. 1 Read pages 329 - 350 (Sections IX.1 to IX.3)
   2 See Instructor for Study Quiz 25

Note: It may be desirable to have the student do parts of these chapters earlier in the course.