SIMON FRASER UNIVERSITY

S.83-91

MEMORANDUM

To: SENATE.	From: SENATE COMMITTEE ON UNDERGRADUATE STUDIES.
Subject: CHANGES - COMPUTING SCIENCE.	Date: 17 NOVEMBER 1983.

Action undertaken by the Senate Committee on Undergraduate Studies at its meeting of November 15, 1983 gives rise to the following motion:

MOTION: "That Senate approve and recommend approval to the Board of Governors, as set forth in S.83-91 the changes in Computing Science, including:-

i) New courses:-

CMPT 100-3 - Structured Basic Programming and Software Packages for Business Administration Students

CMPT 102-3 - Introduction to Programming for Science Students

ii) Prerequisite changes resulting from these new courses"

Subject to the approval of CMPT 100-3 by Senate and the Board of Governors the Senate Committee on Undergraduate Studies approved waiver of the normal two-semester time lag requirement in order that this course may be first offered in Summer 84-2.

SIMON FRASER UNIVERSITY

MEMORANDUM

SCUS 83-57

To: Mr. H.M. Evans Registrar and Secretary to the Senate Committee on Undergraduate Studies	From: Janet Blanchet Secretary, Faculty of I.D.S. Undergraduate Curriculum Committee
Subject: NEW COURSE PROPOSALS, CMPT 100-3, Structured Basic	Date: October 17, 1983

Programming and Software packages
for Business Administration Students.
CMPT 102-3, Introduction to Programming
for Science Students (ISC 83-23 (a)

At a meeting of the Faculty of Interdisciplinary Studies Undergraduate Curriculum Committee held on Tuesday, October 11, 1983 members of the Committee approved the above-noted new courses for inclusion in the Computing Science Calendar entry.

Would you please place this item on the next agenda of the Senate Committee on Undergraduate Studies for consideration.

JB/rj

J. Blanchet

SIMON FRASER UNIVERSITY

MEMORANDUM

To: Dr. T.W. Calvert, Dean
Faculty of
Interdisciplinary Studies

From: Brian Funt, Director
Undergraduate Program
Computing Science

Subject: PREQUISITE CHANGES RESULTING FROM THE ADDITION OF CMPT 100-3 and CMPT 102-3

Date: October 6, 1983

With the addition of CMPT 100-3 (STRUCTURED PROGRAMMING AND SOFTWARE PACKAGES FOR BUSINESS ADMINISTRATION STUDENTS) and CMPT 102-3 (INTRODUCTION TO PROGRAMMING FOR SCIENCE STUDENTS) the following changes in prequisites are required. The intent of these changes is that students with credit for CMPT 100 will be ineligible to proceed further with computing courses unless they take either of CMPT 101 or 103 for no extra course credit. Students with credit for CMPT 102 who wish to proceed further with computing science will be required to take CMPT 104-1.

CMPT 101-4 from: Students with credit for CMPT 103 may not receive further credit for CMPT 101

to: Students with credit for CMPT 100, 102, 103 may not receive further credit for CMPT 101.

CMPT 103-4 from: Students with credit for CMPT 101, 102, or 103 may not take CMPT 103 for further credit.

to: Students with credit for CMPT 100, 101, 102, or 103 may not receive further credit for CMPT 103.

(point of information: CMPT 103 was previously offered as a three credit course and this is why it is included in the list).

CMPT 104-1 from: CMPT 101 (or 103 with a grade of B or higher). The student must select a different language from that studied in CMPT 101 or 103.

to: CMPT 101 (or either of 102 or 103 with a grade of B or higher). The student must select a different language from that studied previously.

CMPT 105-3 from: CMPT 101 (or 103 with a grade of B or higher). Students with credit for CMPT 100 may not take this course for further credit.

to: CMPT 101 or 104 (or 103 with a grade of B or higher).

CMPT 118-3 from: CMPT 101 (or CMPT 103 with a grade of B or higher).

to: CMPT 101 or 104 (or 103 with a grade of B or higher).

CMPT 250-3 from: CMPT 103, 118, at least two appropriate courses in the Environmental Sciences.

to: CMPT 118 and at least two appropriate courses in the Environmental Sciences.

CMPT 340-3 from: Completion of 60 credits including CMPT 103 and two appropriate courses in the Life Sciences (BISC, KIN., PSYC)

to: Completion of 60 credits including CMPT 101 (or 102, 103, or 104 with a grade of B or higher) and two appropriate courses in the Life Sciences (BISC, KIN., PSYC)

SENATE COMMITTEE ON UNDERGRADUATE STUDIES

NEW COURSE PROPOSAL FORM

1. Calendar Information

Department: COMPUTING SCIENCE

Abbreviation Code: CMPT Course Number: 100 Credit Hours: 3 Vector: 3-1-0

Title of Course: STRUCTURED BASIC PROGRAMMING AND SOFTWARE PACKAGES FOR BUSINESS ADMINISTRATION STUDENTS

Calendar Description of Course:

This course is designed to introduce the student to the fundamentals of computer operation and computer programming. The student will learn the basic steps in entering, saving, retrieving, editing and running programs using WATERLOO BASIC on MTS. The class will explore the techniques used in designing and implementing simple computer programs. In addition, the student will have the opportunity to run user-oriented business software packages.

Nature of Course LECTURE AND TUTORIAL

Prerequisites (or special instructions):

B.C. High School Grade 12 Mathematics, or MATH 100.
Students with credit for CMPT 101, 102 or 103 cannot take CMPT 100 for credit.

What course (courses), if any, is being dropped from the calendar if this course is approved: NONE

2. Scheduling

How frequently will the course be offered? EVERY SEMESTER

Semester in which the course will first be offered? 84-2

Which of your present faculty would be available to make the proposed offering possible? S. CAPLIN and G. BIAGIONI

3. Objectives of the Course

To meet the Faculty of Business Administration's computer programming and computer literary requirements for their students.

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: October 5, 1983 20 Oct 83

Department Chairman | Dean | Chairman, SCUS

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

***** Proposal for CMPT 100 ***

SPECIAL SECTION IN BASIC PROGRAMMING AND SOFTWARE PACKAGES FOR BUSINESS ADMINISTRATION STUDENTS

PREREQUISITE: B.C. high school grade 12 mathematics or MATH 100.

Please note that CMPT 100 is designed for those students wishing to satisfy Business Administration requirments in computing science. Students who are planning to take other courses in Computing Science should register in CMPT 101 or one of the other sections normally offered in CMPT 103.

Unlike CMPT 101 and the other sections of CMPT 103 normally offered, CMPT 100 is NOT designed as a rigorous introduction to computer programming, but rather as an introduction to general computer use.

This course is designed to introduce the student to the fundamentals of computer operation and computer programming. The student will learn the basic steps in entering, saving, retrieving, editing and running programs using Waterloo BASIC on MTS. The class will explore the techniques used in designing and implementing simple computer programs. In addition, the student will have the opportunity to run user oriented business software packages.

Upon completion of the course the student will be able to:

a. communicate with the computer using BASIC and fundamental system commands. (eg. let, read, print, LOAD, SAVE, RUN)

b. define concepts such as hardware and software, file systems, etc ...

c. write a computer program in structured BASIC

d. work with selected business software packages

COURSE SCHEDULE

WEEK	TOPIC	MATERIAL COVERED	ASSIGN.	TEST
1	Getting Acquainted With the Computer	what is a computer, organization of a computer, computer systems		
2	Learning to Communicate With the Computer	operating systems, assemblers, compilers, interpreters, computer programs	1	
3	Fundamentals of Basic	constants, variables, numbers, characters, keywords, line numbers, statements	2	
4	Unformatted Input/Output	READ, DATA, RESTORE, INPUT	3	
5	Introduction to Control Statements	GO TO, IF-ENDIF, IF-ELSE FOR/NEXT		MIDTERM 1
6	Additional Control Statements	WHILE-ENDLOOP, LOOP-ENDLOOP LOOP-UNTIL	4	
7	String Manipulation	subscripts, applications		
8	One Dimensional Arrays	defining, subscripts, applications	5	
9	Functions and Subprograms	built-in functions, user defined functions and subprograms		MIDTERM 2
10	Formatted Input/Output	PRINT USING, INPUT USING, images	6	
11	Data File Handling	reading and writing records		
12	Business Application Programs	inventory, interest loans,	7	
13	Review		8	

WEIGHT DISTRIBUTION

Assignments	30%
~~Quiz~~ 1 and 2	30%
Final exam	40%

melton

CMPT 100 will be conducted with three lectures per week - one of which will be in tutorial form. Each major topic is covered by a reading assignment followed by a programming assignment. The CMPT 101/103 open laboratory will not be available for this section of CMPT 103.

Required Materials:

BASIC 1) APPLIED BASIC PROGRAMMING
R. Ageloff, R. Mojena
Wadsworth

2) WATERLOO BASIC - A STRUCTURED PROGRAMMING APPROACH
J.W. Graham, J.W. Welch, K.I. McPhee
Watfac publications

3) INTRODUCTION TO MTS AT SFU (SFU-13)

The above materials are available from the SFU Bookstore.

SIMON FRASER UNIVERSITY

MEMORANDUM

To: Brian Funt, Director
Computing Science, Undergraduate Curriculum Committee

From: R. G. Wyckham, Director
Undergraduate Programs

Subject: New course proposal CMPT 100

Date: October 19, 1983

Faculty of Business Administration fully endorses this proposed course. We have worked with Computing Science over the last several years in designing an Introductory Computing Course for our incoming students. The previous CMPT 101 course we felt was inappropriate to our needs. Most significantly our typical business student has little need for specialized high level languages. Rather he/she should be competent in accessing an interactive terminal (typically found in the marketplace today), writing small computer programs to carry out analytical studies and utilizing pre-canned programs of use in business analysis. Many downstream courses in Business Administration rely heavily on such pre-canned programs.

The proposed CMPT 100 we feel does satisfy these requirements and better equips our students for subsequent courses within the Business Administration curriculum.

SENATE COMMITTEE ON UNDERGRADUATE STUDIES

NEW COURSE PROPOSAL FORM

1 Calendar Information

Department: COMPUTING SCIENCE

Abbreviation Code: CMPT Course Number: 102 Credit Hours: 3 Vector: 3-1-0

Title of Course: INTRODUCTION TO PROGRAMMING FOR SCIENCE STUDENTS

Calendar Description of Course: A programming course which will provide the Science student with a working knowledge of the Fortran language and an introduction to computing concepts, structured programming, and modular design. The student will also gain some knowledge of the Fortran computing environment including the use of numerical algorithm packages.

Nature of Course Lecture and Tutorial

Prerequisites (or special instructions):

Corequisite
MATH 152 or MATH 155 (or MATH 158)
Students with credit for CMPT 100, 101 or 103 cannot take CMPT 102 for credit.
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? Once per year.

Semester in which the course will first be offered? 84-3

Which of your present faculty would be available to make the proposed offering possible? Dr. M.S. Drew

3. Objectives of the Course

The course will provide sufficient background to be useful as the only computing course in the student's career. In addition, the course will provide a solid preparation for further courses which deal in more depth with computational methods and numerical analysis.

4. Budgetary and Space Requirements (for information only)

What additional resources will be required in the following areas:

Faculty

Staff

Library

Audio Visual NONE.

Space

Equipment

5. Approval

Date: October 6, 1983 20 Oct 83

Department Chairman | Dean | Chairman, SCUS

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

CMPT 102-3 Introduction to Programming for Science Students

Intended Audience: All Undergraduates in the Faculty of Science at SFU

Corequisite: Math 152 or Math 155 (or Math 158)

Course Structure: 2 hours lecture and 1 hour tutorial per week, plus ample office hours.

Course Materials: (i) Text - a textbook emphasizing structured programming in FORTRAN 77, with examples drawn from the Sciences. Probable choice -- "Structured FORTRAN 77 for Scientists and Engineers" by D.M. Etter, Benjamin/Cummings Publishing, 1983.
(ii) Study Guide - system description; text editor instruction; course guidelines and direction; and detailed instructions on assignments.

Objective: A programming course which will provide the Science student with a working knowledge of the Fortran language and an introduction to computing concepts, structured programming, and modular design. The student will also gain some knowledge of the Fortran computing environment including the use of numerical algorithm packages.

Scope: The course will provide sufficient background to be useful as the only computng course in the student's career. In addition, the course will provide a solid preparation for further courses which deal in more depth with computational methods and numerical analysis.

Content: Emphasis will be placed on programming skills. Concepts on computational methods will be demonstated. Science content will be at the level and depth as outlined in the assignments.

Workload: There will be 8 assignments in total, 4 of which will be one-week assignments and 4 of which will be two-week assignments. There will be one midterm examination and one final examination. The workload will be evenly distributed; no final project will be assigned.

Standards: The level of achievment demanded in CMPT 103 with regard to programming and documentation will be adhered to (but not exceeded). The level of problem-solving skills will correspond approximately to that expected of CMPT 101 students, due in part to the restriction to Science topics.

Facilities: FORTRAN 77 is available on MTS as *FORTRANVS.

Documentation on this IBM product is in part available now and in part on order in the Reference Room of the Centre. Several additional copies of appropriate documentation should be made available in facilities to be used (e.g. Library facility), notwithstanding an excellent summary of syntax in the textbook.

The beginning-student oriented introduction to the MTS system now in preparation for CMPT 103 will be adopted almost verbatim into the CMPT 102 Study Guide.

Content: The following outline details 13 weeks of instruction and is meant to integrate the order of delivery of material in part with the textbook and the order of assignments. There is some flexibility in the specific distribution of topics from week to week.

PROPOSED COURSE SCHEDULE

WEEK	CONTENT	MILESTONE
1.	Introduction to Computing; Algorithms; Flowcharts. general introduction to the computer; computer languages; MTS: $EDIT; compilation; execution; MTS: $RUN; computer systems; algorithms; flowcharts.	
2.	Arithmetic Computations. constants and variables; data types; arithmetic operations; logical IF; WHILE; List-directed input/output to terminal; MTS: $RERUN; program form: introduction; documentation; program form: termination by internal criterion; intrinsic functions: introduction.	ASST.1 (a)Run a given simple graph; (b)Modify program.
3.	I/O; Control(Introduction). list-directed output: conclusion; complete programs; root-finding program (bisection method); list-directed input from files; interactive user input vs. file input; IF-THEN-ENDIF; flowcharts and pseudocode; implementation of WHILE in VS Fortran.	ASST.2 Root-finding (given flowchart).
4.	Control; Errors. ELSE; ELSEIF; program form: trailer value termination; nested WHILE loops; data validation; program form: end-of-data termination; (roundoff error); data types revisited: truncation and mixed mode; interactive user instructions; foolproofing interactive user entries.	ASST.3 Solve 1st order O.D.E.
5.	DO Loops; Format-Directed I/O. DO -- general form; DO -- pseudocode; nested DO loops; application -- controlling erroneous user entries; intrinsic functions; WRITE/FORMAT combinations; READ/FORMAT combinations; data files in MTS vs. standard Fortran77.	ASST.4 Quadratic equation solver.
6.	Arrays; Strings(Introduction). 1-dimensional arrays; DATA; I/O of 1-D arrays; tables; reports; 2-D arrays; multi-dimensional arrays; nature	

of experimental data; data screening; simple data smoothing; representation of characters; simple string I/O; application -- variable formatting.

7. Subprograms.
   program modularity; top-down design; functions; arithmetic statement function; subroutines; sorting; a sort module; common block; block data.

ASST.5
Data screening; application in Biology.

8. Strings.
   operations with character strings; character string subprograms; printer plots; data validation with CHARACTER input; problems on strings.

MIDTERM EXAM.

9. Subprograms and Arrays; Additional Specifications.
   IMPLICIT; PARAMETER; EQUIVALENCE; variable dimensioning; subprograms and arrays; variable dimensioning of arrays in subprograms; a better sorting module (variable N).

ASST.6
Data validation of string input.

10. Additional Types; Using COMPLEX Type.
    DOUBLE PRECISION; COMPLEX; LOGICAL; quadratic equation solver using COMPLEX type; solving a cubic; a search module; a sorting module with flag.

11. Applications; Introduction to Subroutine Packages
    applications of formatting: implied DO loops; graphing; representation of scientific models in Fortran; (truncation error); introduction to Fortran-callable subroutine packages; random number generation.

ASST.7
Cubic equation module; application in Chemistry.

12. Additional Subprogram Facilities; Extended I/O.
    debugging tools; SAVE; INTRINSIC/EXTERNAL; Format extensions; introduction to IMSL and NAG; use of IMSL plotter; introduction to IMSL routine used in assignment 1.

13. Fortran Computing Environment; History; Files.
    Fortran environment; IBM: OS, MTS (*FTN, *OVERDRIVE); microcomputer implementations; Fortran dialects; Fortran history; Fortran IV restrictions; translation to/from Fortran IV; files; internal files: use in data

validation.

ASST.8
Program with several modules; use of subroutine library module; application in Physics.

Assignments: The following synopses of assignments illustrate the type and number of assignments envisaged as appropriate for CMPT 102-3.

1. (a) Access a pre-programmed source program, and run it. The program sends the sine function and selected end-points down to a pre-compiled subroutine which plots it.
(b)Alter the program to produce a graph for the cosine function with different end-points.

Context => plotting

Concepts =>text editor; compiling; ease of use of computer

2. Root Finding. Given a flowchart, produce a program using Newton-Raphson method for finding roots of given curve-fit to experimental data. E.g.,

$$x^{1.37} - 4.5511x + 1.7065 = 0$$

Two runs: user inputs X(initial), Maximum-Iterations.

Context => experimental curve fit

Concepts => WHILE loop, variables, simple I/O,iteration, relative error

3. Solution of O.D.E. - Euler's Method. Solve lst order differential equation; input equation parameters from input file provided; stop via trailer value.

Context => simple first order equation

Concepts => trailer value; data files; echo printing; nested WHILE loop

4. Quadratric equation solver. User input of equation parameters; trap error -- leading coefficient zero.

Context => quadratic equations

Concepts => data validation; if-then-else; interactive programming; user instructions; error messages; end-of-data termination

5. Data screening ; statistics for Biology data. User inputs number of variables and name of each variable. Produce program to read data from file, preceded by format specification of data. Read all data until end-of-file or 99 entries, whichever comes first. For each variable, calculate highest, lowest, number of nonzero cases, mean, and standard deviation. Print results.

Context => experimental data

Concepts => arrays; logical operators; variable formatting; data validation; data counting; character data

6. Data validation of numeric data read in as character strings. Produce program to analyze input for non-numeric data. Flag individual erroneous characters with asterisks on a second line of output.

Context => data validation of input file

Concepts => character string manipulation; error flagging using graphic-type formatting

7. Module: Cubic equation solution. Produce subroutine to solve a general cubic, covering all three possiblities of root types. Explanation (simple) in Study Guide of how the van der Waals equation of state for a real gas can be written as a cubic. Write program which calls module.

Context => Chemistry

Concepts => flexible subprograms; use of mathematical routines in practical applications.

8. Numerical algorithm package; several modules. Write program to calculate two-dimensional profile of vibrating string using method outlined in Study Guide. Call sophisticated plotting routine for presentation of results in graphic form.

Context => Physics

Concepts => debugging modular programs; scientific programming; interchangable modules; top-down design; Fortran-callable numerical subroutine packages.

SIMON FRASER UNIVERSITY

MEMORANDUM

To: B. Funt, Chairman
Dept. of Compt. Science
Undergraduate Studies Ctte.

From: A. G. Sherwood, Chairman
Faculty of Science
Undergraduate Studies Ctte.

Subject: CMPT 102-3

Date: October 20, 1983

The recommendation to include the above course among the requirements for Honors and Majors degrees in Chemistry, Biochemistry and Mathematical Physics has been made to the Faculty of Science by the department and committees responsible for these programs.

The course looks good and members of the Faculty of Science look forward to cooperating with Dr. Drew in the continuing development of the course, in advertising it among science students in programmes for which the course will not be a requirement, and in encouraging our students to make use of the skills in computing which they will acquire.

I very much appreciate the cooperation of the Computing Science people in the development of the course which should contribute much to the quality of our undergraduate programs.

A. G. Sherwood

/rcw

cc: Dean Calvert
cc: Dean Cochran