

Division: ACADEMIC

 DATE: March 12, 1992

 B: Department: SCIENCE & MATHEMATICS

New Course:

 Revision of Course  
 Information form: XX

 DATED: September 1983 OCT 22, 1985

C: <u>CMPT 220</u>	D: <u>Computer Organization and Assembly Language Programming</u>	E: <u>3</u>
<u>Subject &amp; Course No.</u>	<u>Descriptive Title</u>	<u>Semester Credit</u>

**F: Calendar Description**  
 (THIS COURSE IS THE SAME AS CIS 450). This course is an introduction to computer organization and assembly language programming. The computer organization component of the course concentrates on the components of the central processing unit (CPU) such as: gates, registers, arithmetic logic unit, and memory. The Control Unit of the CPU is introduced by considering the fetch-decode-execute cycle and the data transfer between registers. The assembly language programming component considers machine independent data representation, memory addressing and the software components of the operating system such as the assembler, the linker, and the loader.

**Summary of Revisions:**  
 (Enter date & section)

1985 10 22	1991 09 27
Section:	Section:
F - reworded	D - reworded
O - reworded	F - reworded
P - reworded	H - include CIS courses
R - 2 tests as opposed to 1 midterm	N - new text
	O - reworded
	P - reworded
	R - percent variation

**G: Type of Instruction: Hours Per Week/Per Semester**

Lecture	<u>4</u>	Hrs.
Laboratory	_____	Hrs.
Seminar	_____	Hrs.
Clinical Experience	_____	Hrs.
Field Experience	_____	Hrs.
Practicum	_____	Hrs.
Shop	_____	Hrs.
Studio	_____	Hrs.
Student Directed Learning	_____	Hrs.
Other	<u>5</u>	Hrs.
<b>TOTAL</b>	<b><u>9</u></b>	<b>HOURS</b>

**H: Course Prerequisites:**  
 CMPT 110 OR CIS 110 and any one of:  
 CIS210/230/240/250

**I: Course Corequisites:**  
 NONE

**J: Course for which this course is a pre-requisite**  
 CMPT 310/330

**K: Maximum Class Size:**  
 35

**L: College Credit Transfer** XX  
 College Credit Non-Transfer \_\_\_\_\_

**M: Transfer Credit:**  
 Requested XX  
 Granted \_\_\_\_\_  
 Specify Course Equivalents or Unassigned Credit as Appropriate

U.B.C.  
 S.F.U.  
 U. Vic.  
 OTHER:

*[Signature]*  
 COURSE DESIGNER(S)

*[Signature]*  
 DIRECTOR/CHAIRPERSON

*[Signature]*  
 DIVISIONAL DEAN

*[Signature]*  
 REGISTRAR

AUG 27 1999

220 CONTACTED DEP'T ASSISTANT  
CMPT 200 FOR SCIENCE & TECH. WHO CONFIRMED  
THAT ALL OF THEIR COPIES OF CMPT 220  
OUTLINE ALSO HAD CMPT 200 ON 2ND PG

N: Textbooks and materials to be purchased by students  
(Use Bibliographic Form):

Abel, P., IBM PC Assembly Language and Programming, Second Edition, Prentice Hall,  
1991.  
2 x 3½" High density diskettes  
Portfolio

---

Complete Form with Entries Under the Following Headings:

- O. Course Objectives;      P. Course Content;      Q. Method of Instruction;  
R. Course Evaluation

O. Course Objectives:

The student will be able to:

1. design simple digital circuits (the electrical aspects of circuits are not considered);
2. describe the Von Neumann architecture (stored program concept) of a computer;
3. perform basic operations on numbers of varying bases and convert numbers from one base to another;
4. perform complement arithmetic;
5. design and implement programs in an assembly language;
6. demonstrate an understanding of operating system components such as assemblers, macro processors, linkers, loaders, interrupt systems.

P. Course Content:

Hardware and Introduction to the binary numbering system

- gates, combinational circuits, adders, registers, arithmetic logic units, memories
- concept of sequential circuits

Architecture and Introduction to software

- a simple virtual machine
- the assembly and loading process
- macroprocessing
- IBM PC architecture

**P. Course Content (continued....)**

**Data Representation**

- binary, octal, and hexadecimal numbering system and conversion between bases
- representation of integers, sign and magnitude, 1s complement, 2s complement
- representation of characters and strings

**Assembly Language programming**

- instruction set of the IBM PC
- addressing techniques
- development of system routines (I/O, loader, interrupt system)
- assembly programming techniques (simulation of structured constructs, subroutine linkage, parameter passing).

**Q. Method of Instruction**

The theoretical material is covered in lectures. The concepts are reinforced via programming and non-programming assignments.

**R. Course Evaluation**

Assignments (6 - 10)	50 - 65%
Tests (2 at 10 - 15% each)	20 - 30%
Final exam (at 15 - 20%)	<u>15 - 20%</u>
	100%