SAULT COLLEGE OF APPLIED ARTS & TECHNOLOGY

SAULT STE. MARIE, ONTARIO

COURSE OUTLINE

Course Title: MICROCOMPUTER SYSTEMS II

Course No.: CET226-5

Program: COMPUTER ENGINEERING

Semester: FOUR

Date: MAY, 1990

Author: TYCHO BLACK

New:____ Revision: X

APPROVED: J. McGrath 90/05/14

Chairperson Date

CET226

MICROCOMPUTER SYSTEMS II (5 CREDITS)

PHILOSOPHY/GOALS

This course develops a student's ability to use assembly language in a PC environment effectively. It follows an earlier course, CET127 which developed a basic familiarity with the 8088/8086 instruction set, addressing modes and programming.

CET226 expands upon this by developing an understanding of the use of BIOS and DOS interrupts to interact with the keyboard, screen, printer and diskette services. In addition, arithmetic processing, string processing, table searching, sorting, the use of procedures, macros and linking to high-level languages are covered. A second major emphasis, which is reenforced with lab projects, is the use of assembly language to control peripherals at the chip level. A variety of hardware devices will be controlled, including timer/counters, DMA controllers, interrupt controllers and Uarts in a PC environment.

METHOD OF ASSESSMENT:

3 Written Tests	(20% each)	60%
Lab assignments		35%
Quizzes		5%
		100%

TEXTBOOK:

"IBM PC ASSEMBLER LANGUAGE AND PROGRAMMING"

by Peter Abel (Prentice-Hall)

COURSE DURATION:

5 HOURS PER WEEK FOR 1 SEMESTER (15 WEEKS): 75 HOURS

GRADING SCHEME
CET226

1. TESTS

Three written tests will be given at approximately 5 week intervals. Generally one week advance warning will be given for tests.

2. ASSIGNMENTS

Assignments not handed in by the assigned due-date will be penalized by 25% for each week late. All assignments must be individually completed and submitted by each student.

3. ATTENDANCE

Satisfactory attendance in regularly scheduled classes is essential for successful completion of this course. Poor attendance may remove the right to participate in any upgrading activities at the end of the course.

4. GRADING SCHEME

A+	90	-	100%	Outstanding achievement
A	80	-	89%	Above Average achievement
B	70	-	79%	Average Achievement
C	55	-	69%	Satisfactory Achievement

I Incomplete: Course work not complete at Mid-term. Only used at mid-term.

R Repeat

X A temporary grade that is limited to instances where special circumstances have prevented the student from completing objectives by the end of the semester. An X grade must be authorized by the Chairman. It reverts to an R if not upgraded in an agreed-upon time, less than 120 days.

4. UPGRADING OF INCOMPLETES

When a student's course work is incomplete or final grade is below 55%, there is the possibility of upgrading to a pass when the student's performance warrants it. Attendance and assignment completion will have a bearing on whether upgrading will be allowed. A failing grade on all tests will remove the option of any upgrading and an R grade will result. The highest grade on re-written tests or assignments will be 56%.

Where a student's overall performance has been consistently unsatisfactory, an R grade may be assigned without the option of make-up work.

SPECIFIC OBJECTIVES

The student will demonstrate a knowledge of, and competency in the application of the following topics to problem solving:

BLOCK 1: ASSEMBLY LANGUAGE FUNDAMENTALS AND REVIEW

2 weeks

1. Understand the instruction set and addressing modes of the 8088. (review)
2. Understand the assembly and execution requirements of the Microsoft Assembler and the use of Codeview as a debugger.
3. Understand special program initialization requirements when using the Microsoft Assembler.
4. Understand the differences between COM and EXE program files.
5. Be able to use structured program constructs and good programming style in an assembly language environment.
6. Know the appropriate use of Pseudo-ops in assembly language programs.

BLOCK 2: SCREEN AND KEYBOARD PROCESSING (CHAP. 8,9)

2 weeks

1. Use DOS and BIOS interrupts to display information on the screen and accept input from the keyboard.

BLOCK 3: ADVANCED PROGRAMMING (CHAP. 11,12,13,20,21)

4 weeks

1. Write programs that demonstrate a knowledge of string processing, arithmetic processing, table searching and sorting, as well as other important applications of assembly language.
2. Be able to write and use assembler macros and access them from macro libraries.
3. Be able to link to external procedures and use various mechanisms to pass parameters between them.
4. Be able to link assembler modules to high-level language modules.

BLOCK 4: DISK I/O and PRINTER CONTROL (CHAP. 16,17,18,19)

3 weeks

1. Understand diskette organization in a PC environment.
2. Be able to use File Control Blocks for disk processing.
3. Be able to use File Handles for disk processing.
4. Be able to use BIOS functions to create and read disk files.
5. Understand the requirements for printing in assembler language.

BLOCK 5: PERIPHERAL DEVICES AND PROGRAMMING

4 weeks

1. This block will involve the study and control of various devices using PC's and the MATs (Microcomputer Applications Trainers). A variety of programmable devices including the 8253 Programmable Interval Timer, the 8259 Programmable Interrupt Controller, the 8255 PPI ,the 8237 Direct Memory Access Controller, the 8250 Uart, A/D and D/A converters and other support chips will be studied. The student is expected to understand the architecture, registers, programming requirements and application of such chips in PC systems.

2. Be able to describe the interface requirements and operation of the above-named chips in computer systems, like the IBM-PC. The student will be expected to describe the operation of these devices at the schematic level, especially where implemented in MAT, URDA or PC systems in lab projects or discussions.

NOTE: The times shown as the expected duration of each block are approximate and may vary as circumstances require.