SAULT COLLEGE OF APPLIED ARTS & TECHNOLOGY

SAULT STE. MARIE, ONTARIO

COURSE OUTLINE

Course Title: COMPUTER SYSTEMS I

Course No.: CET 200

Program: COMPUTER ENGINEERING TECHNOLOGY

Semester: THREE

Date: AUGUST, 1986

Author: PETER SAVICH

New: Revision: X

APPROVED: Chairperson Date 86-08-29

CET 200

COMPUTER SYSTEMS I

PHILOSOPHY/ GOALS

The objectives of this course are to provide the student with a knowledge of the PDP-11 family of computers and the practical experience of programming in assembly language the PDP-11 computers. The student will become proficient in writing assembly programs using the editor available for the multi-tasking operating system RSX-11M. The student should then be capable of assembling, linking, running and debugging the programs. Subsequent advanced courses will provide the student with even more knowledge and experience with the PDP-11 family of computers.

COURSE OBJECTIVES/ OUTLINE

BLOCK 1. System Architecture and Machine Language Programming

At the end of this block the student should be capable of:

1. Describing the hardware components of a computer system using block diagrams.

2. Discussing the architecture of the PDP-11 family of computers using schematics.

3. Describing the operation and use of all the addressing modes available to the PDP-11 family of computers.

4. Highlighting the various operating systems designed for use on PDP-11 family of computers and the advantages/disadvantages each Op/Sys has.

5. Writing and discussing the intention of any simple instruction from the Macro-11 assembly language instruction set.

6. Editing, asembling, linking, running, and debugging simple programs.

7. Using the ODT debugging technique to set breakpoints, alter register contents, memory address contents, single-step through a program, and utilize relocation registers.

8. Converting from mneumonics to machine code octal instructions.

BLOCK 2. Subroutines, Macros and Libraries for Programming

At the end of this block the student should gain some valuable experience in the following areas:

1. Using subroutines and macros to modularize more complex programs.

2. Linking and mapping of seperate files for the running of programs.

3. Using the librarian to create and maintain a library of routines.

4. Describing the various transmission methods for passing of arguements to a subroutine.

BLOCK 3. File Handling

At the end of this block the student should have experience with the file control services (FCS) software available from Digital and be capable of:

1. Describing the principles of file control using schematics and block diagrams.

2. Using the FCS macros to perform Input/Output operations.

3. Utilizing the FCS supplied macros to write programs that manipulate files that contain data that must be transformed or processed before transfer to the terminal, printer, or another file.

BLOCK 4. Floating Point Numbers

At the end of this block the student should be capable of:

1. Converting between decimal representation of fractions and integers into the floating point representation and vice versa.

2. Using the floating point instruction set in a programming environment.

EVALUATION PROCEDURES AND ATTENDANCE REQUIREMENTS

The student will be assessed through a series of written tests, practical demonstrations, quizzes, attendance, and programming assignments.

All tests and assignments will be conducted on a timely basis with one week notice.

All programming assignments submitted past the due date will be accepted but 50% of mark earned will be recorded. (50% PENALTY)

All assignments will be weighted to equal value.

All quizzes will be equivalent in weight to that of assignments and may not be given any advance notice.

Attendance in all labs and classes is expected. A bonus program of up to 10% of final mark will be given to any student with a perfect record of attendance. A student with 80% attendance or less will not receive any bonus.

RECORD OF ATTENDANCE	BONUS
~~less than~~ 80% and less.	0%
81% - 85%	2%
86% - 90%	5%
91% - 95%	7%
96% - 100%	10%

Re-writes will be available for the written tests. However, regardless of mark earned on the re-write (assuming the mark was above 65%) a maximum mark of 65% will be recorded for that particular test.

The final mark will be calculated using the following formula that uses the raw marks recorded from the assessment techniques described earlier.

written tests	60%
programming assignments	20%
quizzes	10%
practical demonstrations	10%
FINAL MARK	100%
attendance BONUS	10%

The grade system to be used is as follows:

PERCENTAGE MARK	GRADE MARK
80% - 100%	A
66% - 79%	B
55% - 65%	C
below 55%	R

The final grades must be submitted by Jan 3/87. Students must have a valid reason in order to receive an "X" grade if any work remains uncompleted at this deadline. DO NOT assume you will be given the "X" grade unless you have been told specifically.