

SAULT COLLEGE OF APPLIED ARTS & TECHNOLOGY

SAULT STE. MARIE, ONTARIO

COURSE OUTLINE

Course Outline: PROGRAMMING LANGUAGES

Code No.: EDP318-4

Program: PROGRAMMER AND PROGRAMMER/ANALYST

Semester: FOUR/SIX

Date: JANUARY 1987

Author: DENNIS OCHOSKI

New: _____ Revision: X

APPROVED:


Chairperson

86-12-17
Date

PROGRAMMING LANGUAGES

EDP318-4

Length of Course: 5 periods per week for one semester

Texts: IBM PC/8088 Assembly Language Programming - Avtar Singh and
Walter A. Triebel
VAX Fortran - Charlotte Middlebrooks

Purpose: This is a two-part course designed to familiarize the student with the more technical aspects (architecture) of a computer, at the same time teaching the student two new programming languages.

The Assembly language will be used to give the student a closer insight into the functioning of a computer (in addition to learning Assembler) whereas Fortran will give the student knowledge of an additional higher level language used extensively in various data processing environments.

Objectives: When this course is completed, the student should be able to:

- a) understand the characteristics of the binary and hexadecimal numbering systems,
- b) understand the functioning of the 8088 microprocessor,
- c) code assembler language programs to handle the screen, perform arithmetic, and, perform table searches and sorts,
- d) trace machine execution to aid in debugging programs (have the ability to read "dumps" of memory locations)
- e) understand the basis of Fortran programming and in which environments it is used,
- f) write, store, compile, link, and run Fortran programs,
- g) create, access, and update Fortran files.

Student Evaluation

The student's final grade will consist of the following components:

Tests - Assembler (2 x 15)	30%	<u>Grading:</u> A -- 85 to 100%
- Fortran (2 x 15)	30%	
Assignments - Assembler (2 x 10)	20%	B -- 70 to 84
- Fortran (2 x 10)	20%	C -- 60 to 69
	-----	R -- 0 to 59
	100%	

Assignment Deadlines: Each assignment must be handed in ON **TIME**, otherwise they are subject to a 10% deduction per day late.

All assignments must be handed in, otherwise the student has not fully completed the course and is subject to receiving an "R" grade.

NOTE: There will be no re-write in this course.

Material To Be Covered

PART A: Assembler

<u>TOPIC</u>	<u>DESCRIPTION</u>	<u>REFERENCE</u>
1	<u>The 8088 Microprocessor</u> <ul style="list-style-type: none">- the IBM PC- architecture of the IBM PC- programming with assembler on the IBM PC	Singh/Triebel: chapter 1 Lecture notes
2	<u>The Binary and Hexadecimal Numbering Systems</u> <ul style="list-style-type: none">- representing data- number system conversion- performing arithmetic- machine execution	Singh/Triebel: chapter 2 Lecture notes
3	<u>The Assembler Language</u> <ul style="list-style-type: none">- introduction- segment, data, and pointer registers- DEBUG, TRACE, and GO commands- reading the contents of memory- steps in developing a program<ul style="list-style-type: none">- source statements- assembler instructions- how to edit, assemble, and run a program- data definition- program logic	Singh/Triebel: chapters 3,4,5 Lecture notes
4	<u>The 8088 Instruction Set</u> <ul style="list-style-type: none">- addressing modes- instruction types- data transfer instructions- arithmetic instructions	Singh/Triebel: chapters 6,7 Lecture notes
5	<u>Table Processing</u> <ul style="list-style-type: none">- table definition- table searching	Singh/Triebel: chapter 7 Lecture notes

PART D: FORTRAN

<u>TOPIC</u>	<u>DESCRIPTION</u>	<u>REFERENCE</u>
1	<u>Introduction</u> - Fortran programming overview - output labels - constants, variables, and symbolic names - arithmetic expressions	Middlebrooks: chapters 1,2
2	<u>Input/Output Files</u> - READ statement - WRITE statement - VAX files	Middlebrooks: chapter 3
3	<u>Components of Fortran Programming</u> - Fortran syntax - DO Loops - numeric and character data - formatting input and output - conditional statements - structured Loops	Middlebrooks: chapters 4,5,6,7,8
4	<u>One-Dimensional Arrays</u> - definition of an array - using the DO Loop with arrays - manipulating arrays - applications using arrays	Middlebrooks: chapters 9,10