SAULT COLLEGE OF APPLIED ARTS & TECHNOLOGY

SAULT STE. MARIE, ONTARIO

COURSE OUTLINE

COURSE TITLE: PROGRAMMING LANGUAGES

CODE NO.: EDP318-6   **SEMESTER:** TWO

PROGRAM: COMPUTER PROGRAMMER

AUTHOR: DENNIS OCHOSKI

DATE: JANUARY 1995

PREVIOUS OUTLINE DATED: JANUARY, 1994

New: _____   Revision: ___X___

APPROVED: _____   _____
   DEAN, SCHOOL OF BUSINESS &   **DATE**
          HOSPITALITY

PROGRAMMING LANGUAGES	EDP318
COURSE NAME	**COURSE CODE**

LENGTH OF COURSE: Five periods per week for one semester

TEXTS: <u>Vax-Basic</u> <u>with</u> <u>Structural</u> <u>Problem</u> <u>Solving,</u> 2nd Edition by David G. Weinman and Barbara L. Kurshan
<u>Structured</u> <u>Cobol</u> <u>Programming,</u> 7th Edition by Stern and Stern

PURPOSE: This course will provide students with an opportunity to develop their computer skills by introducing them to program development and structured programming techniques. This exposure will provide a foundation for more advanced study in the following semesters.

The course involves theory related to programming and application of this theory will be implemented using two high level programming languages.

COURSE OBJECTIVES:

1. learn problem-solving techniques and apply them to programming

2. learn and understand the program development process

3. develop logic skills

4. develop solutions using the Basic programming language

5. develop solutions using the Cobol programming language

PROGRAMMING LANGUAGES   EDP318

PART 'A':   The following modules pertain to program development and
            structured programming techniques.
            (introduction and lecture notes)

Module 1:

Objectives:

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

1) differentiate between high level languages and machine language

2) understand the basic make-up of a computer program and state its
    purpose

3) explain why there is a need for custom-designed programs

4) list the steps involved in the program development process

5) understand the main problem of computer programming addressed by
    structured programming

6) state the meanings of program "reliability" and "readability"

7) explain how a hierarchy chart relates to problem-solving

8) state the purpose of pseudo code and why it is useful

9) state the difference between an interpreter and a compiler

10) differentiate between syntax errors and logic errors

11) differentiate between a source program and an object program
    (compiling, linking, and running a program)

12) explain the importance of program documentation

PART 'B': The following modules pertain specifically to the Basic programming language and expand on the concepts of the language discussed in the first semester.

Module 1: **Review of the Basic Programming Language** introduced in the first semester
(chapters 1, 2, 3)

Objectives:

When this module is completed, the student should be able to recall:

1. Basic coding conventions

2. the use of data types (numeric, alphanumeric)

3. the use of arithmetic expressions

4. input and output statements

5. the concept of variables and constants

6. the use of conditional statements

Module 2: **More Advanced Conditional Statements and Program Looping**
(chapters 4, 5, 6)

Objectives:

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

1. implement conditional statements such as the IF and IF-THEN-ELSE and SELECT/CASE statements

Module 2: (cont'd)

2. use logical and relational operators to check for specific
    conditions
3. define the term "collating sequence"
4. apply the concept of looping using the various forms of looping
    structures such as the FOR/NEXT, WHILE/NEXT and UNTIL statements
5. use the PRINT USING statement to format output

Module 3:   **Modular Programming**
            (chapter 6)

Objectives:

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

1. apply the use of sub-routines to incorporate structural
    programming

2. use structure charts to determine how programs should be
    modularized

3. list several advantages of modularizing programs

MODULE 4:   **Arrays**
            (chapter 7 – pages 136–151)

Objectives:

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

1. explain and use subscripts

2. use the DIM statement to define arrays

3. manipulate one-dimensional arrays

5. create, read from and write to a sequential file

PART 'C'   The following modules pertain specifically to the Cobol
programming language

Module 1:   Background and Introduction to Cobol

Objectives:

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

1. understand the purpose and origin of the COBOL programming
    language

2. understand the basic structure of a COBOL program

3. understand COBOL's general coding conventions and format rules

Module 2:   Basic Cobol Elements Relating to the Divisions Within a Cobol Program and Data Definition

Objectives:

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

1. identify the four divisions of a COBOL program, the A and B
    margins, comment lines, and paragraph names

2. describe the purpose of and code of the Identification Division and the Environment Division
3. understand the rules for forming data-names and constants in COBOL
4. describe how input and output files are defined and described in the Data Division
5. understand how storage can be reserved for fields not part of input or output such as constants and work areas

Module 3:   **Development of the Procedure Division to Process Data**

Objectives:

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

1. access input and output files

2. understand MOVE operations and the various options of the MOVE statement

3. understand the flow of control from one module to another using the various options of the PERFORM statement

4. understand the ways in which arithmetic may be performed in COBOL

5. edit output for report purposes

6. use the IF statement options for processing conditions

PROGRAMMING LANGUAGES   **EDP318**

STUDENT EVALUATION:

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

```
Tests   -  Basic  (2 x 15%)    30%
        -  COBOL  (2 x 15%)    30%


Assignments - Basic 54 x 5%) 25%
            - COBO (3 x 5%)  15%

                              _____
                               100%
```

```
A+    90-100%
A     80- 89%
B     70- 79%
C     60- 69%
R     REPEAT - under 60%
```

SPECIAL NOTES:

1. Students with special needs due to such things as physical
    limitations, visual impairments, hearing impairments or learning
    disabilities are encouraged to discuss required accommodations,
    confidentially, with the instructor.

2. There will be no re-writes in this course except in situations out
    of the control of the student (such as illness, urgent family
    matters, etc.) in which a re-write may be issued at the discretion
    of the instructor.

3. Assignments received after the due date are subject to a grade of
    zero except in situations as specified in #2 above.