

SAULT COLLEGE OF APPLIED ARTS & TECHNOLOGY

SAULT STE. MARIE, ONTARIO

COURSE OUTLINE

Course Outline: ADVANCED STRUCTURED COBOL

Code No.: EDP 223

Program: _____

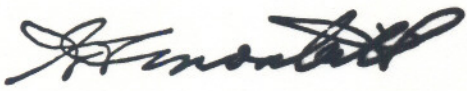
Semester: ONE

Date: SEPTEMBER, 1985

Author: G. Wied

New: _____ Revision: X

APPROVED:


Chairperson

85-09-04
Date

ADVANCED STRUCTURED COBOL

EDP 223

Course Name

Course Number

OBJECTIVES:

To teach the student the structured design methodologies based upon top-down design, functional decomposition and pseudocode.

To provide the student with a firm foundation in the concepts and techniques of structured program design and structured COBOL programming. The student should be capable of solving a wide variety of business-type problems using COBOL.

To challenge the student with regards to understanding, using, relating and integrating concepts learned.

To stress the importance of the group process in the learning situation.

To further motivate the student to research and learn independently.

STUDENT PERFORMANCE REQUIREMENTS:

At the conclusion of each section the student will be required to complete a programming assignment. These assignments become increasingly complex and require more and more creative thinking and applicable design in order to arrive at an acceptable solution and output.

The student must learn to plan and design carefully so as to avoid imprudent use of the computer time. The student is required to desk check programs and participate in the design and code walk-throughs as indicated to ensure this. In order to participate the student must have hierarchy charts, pseudocode, or listings available as required.

The student is required to interact positively in a group process. Group criticism, suggestion or praise must be accepted with equal ease and a helpful constructive attitude must be developed for enacting and re-enforcing group learning.

The student may additionally be required to do individual or group research, file building, documentation, explanation of process etc. to further facilitate learning.

EVALUATION METHOD:

Tests	60%
Assignments & Participation	40%
*Final	60%

Grading	A -- 85 - 100%
	B -- 75 - 84%
	C -- 60 - 74%
	I -- 40 - 60%
	R -- Less than 40%

* Students receiving an I grade will write a final exam. Successful completion of the final will convert the I to the appropriate grade. Failure to write and pass the final will convert the I to an R and the student will have to repeat the course.

NOTES:

The assignments are designed to help the student learn by doing. A concerted effort to understand and complete the assignments will be valuable study for the tests.

Assignments must be neat and complete. Documentation requirements and exact format will be handed out separately with each assignment. Programs must be structured.

Assignments must be handed in by the due date. Late assignments will be penalized 20% per day and will not be accepted after 5 days.

TOPIC OUTLINE

1 INTRODUCTION TO STRUCTURED PROGRAMMING

- Historical development of structured programming
- Top-Down design
- The structures...Sequence, Condition, Iteration
- Improved programming technologies...Teams, Walk-Throughs, Support Library
- Use of structured programming by industry
- Sample application
- The tools ... IPO Chart, Hierarchical Chart, Pseudocode
- Comparison of structured vs unstructured

2 NESTED IF STATEMENTS

- Introduction
- When and how to write a Nested If statement
- Combined If statements and Logical Operators
- Complex If statements
- Sample application using the tools

3 CONTROL BREAKS & DATATRIEVE

- Introduction
- Control break processing
- Multiple level control breaks
- Datatrieve

4 TABLE PROCESSING

- Introduction
- Defining lists of related items
- The occurs clause
- Subscripts and Indexes
- Tables - when & when not to use index/subscript
- Table lookup and search, search verb
- Perform varying

MULTIPLE LEVEL TABLE PROCESSING

- Variable length tables
- Two and three level table processing
- Two and three level variable length table processing

5 ADDITIONAL COBOL STATEMENTS

- Corresponding option (move-add-sub)
- Level 66 renames clause
- Condition names with group level items
- Name qualification
- Justified right
- Blank when zero

5 ADDITIONAL COBOL STATEMENTS - (CONT'D)

- Low values and high values
- All figurative constant
- Alter statement
- Accept statement
- Inspect statement
- String & Unstring statement

GOOD PRACTICES AND BAD

THROUGHOUT THE SEMESTER THE FOLLOWING PRACTICES ARE STRESSED:

- use appropriate comments
- eliminate 77 levels entries
- choose meaningful names
- indentation and spacing to facilitate readability
- avoiding commas
- restrict switches and subscript to a single use
- use 88 level entries
- perform paragraphs not sections
- using compute verb for multiple arithmetic operators
- initialize tables dynamically
- avoid constants and literals
- use READ INTO, WRITE FROM
- consider use of Datatrieve
- stub testing