

SAULT COLLEGE OF APPLIED ARTS & TECHNOLOGY

SAULT STE. MARIE, ONTARIO

COURSE OUTLINE

Course Title: PROGRAMMING LANGUAGES  
-----  
Code No.: CET 204  
-----  
Program: COMPUTER ENGINEERING TECHNOLOGY  
-----  
Semester: THIRD  
-----  
Date: FALL 1989  
-----  
Author: F. TURCO  
-----

New: X      Revision: -  
-----

Approved: LP Carozzella  
Chairperson

Date: \_\_\_\_\_

## CET204

### PROGRAMMING LANGUAGES

#### GENERAL OBJECTIVES

The objectives of this course are to provide the student with programming skills in a variety of programming languages. The languages to be used are VAX COBOL and VAX BASIC. The students will do a comparative study of similarities and differences in COBOL, BASIC and FORTRAN (which was previously studied). The students will also be taught structured programming concepts and the programming tools and techniques available in problem solving. The course will take a relatively non-mathematical approach but will include an in-depth study of programming techniques such as modularization, file handling, error handling, string handling, and formatted output. The students will utilize a variety of VMS development tools such as the DEBUGGER and the LIBRARIAN.

#### TEXTBOOKS:

1. "PROGRAMMING USING VAX BASIC" - Wayne Muller
2. "STRUCTURED COBOL PROGRAMMING" - Stern/Stern

#### ASSESSMENT:

Theory Tests, Practical Tests and Quizzes	60%
Assignments	40%

Some minor modifications to the above percentages may be necessary. The instructor reserves the right to adjust the mark up or down 5% based on attendance, participation and whether there is an improving trend.

- \* - All Assignments must be completed satisfactorily to complete this course. Late hand in penalties will be 5% per day. Assignments will not be accepted past one week late unless there are extenuating and legitimate circumstances.

## GRADING SCHEME

### 1. TESTS

Written tests will be conducted as deemed necessary; generally at the end of each block of work. They will be announced about one week in advance. Practical on-line tests will be conducted in which time to complete the assigned problems will be a factor in the evaluation. Quizzes may be conducted without advance warning.

### 2. ASSIGNMENTS

Assignments not completed by the assigned due-date will be penalized by 5% per day late. All assignments must be completed satisfactorily to complete the course.

### 3. GRADING SCHEME

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

U 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 Chairperson. It reverts to an R if not upgraded in an agreed-upon time, less than 120 days.

### 4. UPGRADING OF INCOMPLETE

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.

The method of upgrading is at the discretion of the teacher and may consist of one or more of the following options: assigned make-up work, re-doing assignments, re-writing of tests, or writing a comprehensive supplemental examination.



## BLOCK\_1 LANGUAGE\_FUNDAMENTALS

Languages in general have similarities in both structure and commands. There will be a comparison of how each language handles basic concepts such as defining data, program code, file handling, formatted output. Since the students have previous experience in FORTRAN, the comparative study will evolve from that which they are accustomed to.

At the end of this block students will be able to:

1. Illustrate the COBOL and BASIC approaches to:
  - a) Program Identification
  - b) Commenting
  - c) Data Definition
  - d) File Definition
  - e) Assignment of Data
  - f) Looping Control
  - g) Decision Making
2. Write and develop programs in the BASIC and COBOL environment.

## BLOCK\_2 STRUCTURED\_APPROACH\_TO\_PROBLEM\_SOLVING

Regardless of what language programs are written in, it is important that students prepare good habits when it comes to problem solving. A disciplined approach to problem solving can and will simplify the students effort. Several tools and techniques will be addressed.

At the end of this block the students will be able to:

1. Describe and use the following structured programming tools and techniques:
  - a) Program FlowCharts
  - b) Pseudocode
  - c) Inspections and Walkthroughs
2. Provide thorough and useful program documentation.
3. Produce Modularized and structured code using programming features such as:
  - a) Subroutines
  - b) Functions
  - c) Common Blocks of code
  - d) Passing of Parameters
  - e) Readable code by proper spacing and indentation
4. Use the Library Facilities as well as Debugging.

### BLOCK\_3 FILE\_INPUT,\_OUTPUT\_AND\_ERROR\_HANDLING

This block will introduce the student to the various types of files on the VAX, and the typical ways of using them. The students will also learn to write error handlers to respond to run time errors.

At the end of this block the student shall be able to:

1. Format output using the PRINT USING statement in BASIC and the variety of PICTURE clauses in COBOL.
2. Discuss the characteristics of the following types of file:
  - a) Terminal Format
  - b) Sequential
  - c) Relative
  - d) Indexed
3. Discuss the methods of performing typical file maintenance activities when using the various types of files such as:
  - a) Add data records
  - b) Modify records
  - c) Delete records
4. Describe the methods of using error handlers to deal with run-time errors.
5. Write COBOL and BASIC programs utilizing the various techniques described above.

### BLOCK\_4 ARRAY\_AND\_STRING\_HANDLING

The students have previously been exposed to one and two dimensional arrays as they are used in FORTRAN. They will now use them in the COBOL and BASIC languages. String manipulation is also a common programming required and will be addressed.

At the end of this block the student shall be able to:

1. Discuss the concept of subscripted variables and the use of one and two dimensional arrays.
2. Discuss the common requirements of programs that process string data.
3. Write programs in both languages that utilize array processing and string manipulation.