SAULT COLLEGE OF APPLIED ARTS & TECHNOLOGY

SAULT STE. MARIE, ONTARIO

COURSE OUTLINE

Course Title: **Assembly Language Programming**

Code No.: **CST202**

Program: **COMPUTER ENGINEERING TECHNOLOGY**

Date: **August 1997**

Semester: **Third (3)**

Author: **Professor Gerry Davies**

Previous
Outline Dated: **August1996**

APPROVED:

Dean Date Aug 25/97

Total Credits: **5**

PREREQUISITES: **none**
LENGTH OF COURSE: **4 HOURS PER WEEK for 16 weeks**
TOTAL CREDIT HOURS: **64**

I. Course Description:

This course develops a student's ability to use assembly language in a PC environment effectively. The student will learn the architecture of the Intel family of processors, the instruction set, and the operation of the processor as instructions are executed. The student will develop skills in writing programs in assembly language and in using a variety of software tools such as assemblers, linkers and debuggers. The use of DOS and BIOS calls to program hardware devices will be studied, as well as software techniques including the use of macro and object libraries, and linking assembly routines to higher level languages such as C++.

II. LEARNING OUTCOMES AND ELEMENTS OF PERFORMANCE:

(Generic Skills Learning Outcomes placement on the course outline will be determined and communicated at a later date)

A. Learning Outcomes:

1. Understand the architecture, operation, and instruction set of a modern microprocessor.
2. Be able to write standard structured programming constructs using assembly language.
3. Understand the structure of DOS and BIOS calls and be able to use them to program the keyboard, monitor, printer and disk.
4. Use software tools such as editors, assemblers, linkers and debuggers to develop assembly language programs.
5. Write assembly language programs illustrating standard programming techniques including arithmetic operations, looping, decision-making, array handling and strings.
6. Utilise software development techniques including the use of macro and object libraries and linking assembly language routines to high level language programs.

B. Learning Outcomes with Elements of Performance:

1. Understand the architecture, operation and instruction set of a modern microprocessor.

Elements of the performance:

- Draw a block diagram of the 8088 processor and describe the function of the parts.
- Describe the operation of simple instructions as they are fetched, decoded and executed by the processor.
- Describe the operation of 8088 instructions.
- Test the operation of small programs using debug and the assembler, linker and debugger.
- Describe the ways of defining data in assembly language programs.

Chapters 1 through 4 of the text

2. Be able to write standard structured programming constructs using assembly language.

Elements of the performance:

- Describe the operation of the boolean and comparison instructions and the transfer of control instructions.
- Form the equivalent of high-level structured programming constructions such as if, while, for and case.
- Use the structures in assembly language programs.

Chapter 6 of text

3. Understand the structure of DOS and BIOS calls and be able to use them to program the keyboard, monitor, printer and disk.

Elements of the performance:

- Describe the structure of DOS and BIOS calls, and how to use them in assembly language programs.
- Use DOS INT 21 calls to perform keyboard and monitor I/O.
- Use BIOS INT 10 calls to perform monitor output.
- Use DOS INT 21 functions to perform simple file input and output.

Chapter 5, 11 and 12 of text

4. Use software tools such as editors, assemblers, linkers and debuggers to develop assembly language programs.

Elements of the performance:

- Use the DOS debug program to enter, test, modify and save programs.
- Use the Turbo Assembler, Turbo Linker and Turbo Debugger to test programs.

Chapters 1 and 4 of text

5. Write assembly language programs illustrating standard programming techniques including arithmetic operations, looping, decision-making, array handling and strings.

Elements of the performance:

- Demonstrate the arithmetic capabilities of the 8088 processor.
- Write programs that perform conversions between numeric and string data.
- Describe and use the string-handling instructions of the 8088.
- Write programs using arrays of numeric and string data.

Chapters 7, 8 and 9 of text

6. Utilise software development techniques including the use of macro and object libraries and linking assembly language routines to high level language programs.

Elements of the performance:

Write programs using multiple procedures and files.
Create and use macro and object libraries.
Demonstrate the techniques of passing parameters to external procedures.
Demonstrate the technique of calling assembly language routines from C.

Chapters 9, 10 and 13 of text

III. TOPICS TO BE COVERED:

Note: These topics sometimes overlap several areas of skill development and are not necessarily intended to be explored in isolated learning units or in the order below.

1. Assembly Language Fundamentals And Addressing Modes
2. Debugging Features Of Turbo Debug and Codeview
3. Macros And Macro Libraries
4. Array And String Handling
5. Numbering Systems
6. Interfacing Assembler with C language

V. METHOD OF EVALUATION:

3 THEORY TESTS (20% each)	60%
ASSIGNMENTS	25%
QUIZZES AND PRACTICAL TESTS	15%

TESTS

Written tests will be announced about one week in advance. Quizzes may be conducted without advance warning, but will normally be scheduled about one week in advance.

ASSIGNMENTS AND LAB ACTIVITIES:

Lab activities and assignments represent a very important component of this course in which practical 'hands-on' skills will be developed. Because of this, **lab attendance is expected** and the satisfactory completion of all assignments is required. It is the student's responsibility to discuss absences from regularly scheduled labs with the instructor so that alternate arrangements (where possible) can be made to complete the lab requirements.

It is acceptable that students consult with each other in relation to their assigned problems. However, it is unacceptable to copy programs written by someone else and submit them as your own work

ATTENDANCE:

Absenteeism will affect a student's ability to succeed in this course. Absences due to medical or other unavoidable circumstances must be discussed with the instructor, so that remedial activities can be scheduled. A Quiz or Test missed because of an unauthorized absence may result in a zero grade being assigned.

The following letter grades will be assigned in accordance with the School of Engineering policies:

Course Grading Scheme

A+	90% - 100%	consistently outstanding achievement
A	80% - 89%	outstanding achievement
B	70% - 79%	consistently above average achievement
C	55% - 69%	satisfactory or acceptable achievement in all areas subject to assessment
R	less than 55%	repeat - the student has not achieved the objectives of the course and the course must be repeated
CR		Credit Exemption
S		satisfactory given at midterm only
U		unsatisfactory given at midterm only
W		Withdrawal from a course before the "drop deadline"

X a temporary grade

An 'X' grade is limited to instances where exceptional circumstances have prevented the student from completing objectives by the end of the semester. An "X" grade must be arranged before the deadline for grade submissions and is granted at the discretion of the Professor. The 'X' grade must also have the Dean's approval and has a maximum time limit of 120 days.

UPGRADING OF INCOMPLETES

When a student's course work is incomplete or final grade is below 55%, there is the possibility of upgrading to a pass when a student meets all of the following criteria:

1. The student's attendance has been satisfactory.
2. An overall average of at least 40% has been achieved.
3. The student has not had a failing grade in all of the theory tests taken.
4. The student has made reasonable efforts to participate in class and complete assignments.

Note: A student may be assigned an 'R' grade early in the course for unsatisfactory performance.

VI. REQUIRED STUDENT RESOURCES:

1. TEXTBOOK: **Assembly Language for the IBM-PC**
Kip Ervine
Macmillan

2. DISKETTES: 2 3-1/2" HD Diskettes.

ADDITIONAL RESOURCE MATERIALS AVAILABLE:

MS-DOS Programmers Reference
IBM ROM BIOS Programmer's Quick Reference
IBM DOS Functions Programmer's Quick Reference

Course Name:
Assembly Language Programming

Course No.
CST 202

VI. SPECIAL NOTES

1. All students should be aware of the Special Needs Office in the college. If you have any special needs such as being visually impaired, hearing disabled, physically disabled, learning disabilities you are encouraged to discuss required accommodations confidentially with the Professor and/or contact the Special Needs Office, Room E1204, Ext 493, or 717, or 491 so that support services can be arranged for you.

2 Your Professor reserves the right to modify the course as he/she deems necessary to meet the needs of students.

3. It is the responsibility of the student to retain all course outlines for possible future use in gaining advanced standing at other post-secondary institutions.

4. Plagiarism
Students should refer to the definition of "academic dishonesty" in the "Statement of Student Rights and Responsibilities". Students who engage in 'academic dishonesty" will receive an automatic failure for that submission and/or such other penalty, up to and including expulsion from the course, as may be decided by the professor.

5. Substitute course information is available at the Registrar's office.

VII. PRIOR LEARNING ASSESSMENT:

This course is currently not PLA'ble . The student must take the course in its entirety.