# SAULT COLLEGE OF APPLIED ARTS & TECHNOLOGY

SAULT STE. MARIE, ONTARIO

## COURSE OUTLINE

Course Title: **Interfacing**

Code No.: **CET 302**

Program: **COMPUTER ENGINEERING TECHNOLOGY**

Date: **January 1997**

Semester: **Sixth (6)**

Author: **Professor Peter Savich**

Previous
Outline Dated: **January 1995**

APPROVED: Joseph C Fuchter 97 01 03

Dean Date

**Total Credits:** **5**

**PREREQUISITES:** **CET 127, CET 226, CET 228**

**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 software fundamentals were learnt in two earlier courses. The CET127 course developed a basic familiarity with the 8088/8086 instruction set, addressing modes and programming. The CET226 course expanded upon this by developing an understanding of the use of BIOS and DOS interrupts to interact with the keyboard, screen, printer and diskette services, i.e. the peripheral devices. The CET302 Interfacing course develops the students ability to design and maintain interfaces to peripheral devices in microcomputer systems. The hardware fundamentals learned in the CET 228 Microcomputer Circuits and Applications course will also be augmented in this course with a series of practical lab exercises. Simply put the marriage of software and hardware in the computer world means **interfacing**. In this course we study how to interface, the interfacing standards and the problems with interfacing and learn some of the troubleshooting techniques.

A variety of computer buses will be studied in this Interfacing course such as the ISA, EISA, 80486 system Architecture, PCI System Architecture, Pentium Processor Architecture, and Microchannel Architecture or MCA. Peripheral buses such as SCSI, ESDI, IDE, and the GPIB or IEEE-488 Bus standard will also be studied. In addition, arithmetic processing, string processing, table searching, sorting, the use of procedures, macros and linking to libraries and high-level languages are covered.

## 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. Demonstrate proficiency at writing programs in Assembly Language, utilizing MASM or Turbo Assembler/Debugger, and utilizing assembler macros and macro libraries.

2. Write program modules that interface with High level languages such as "C". The assembly modules will perform arithmetic processing, table searching, and/or sorting. The "C" level language module will perform Input/ Output

3. Interface to the numeric co-processor.

4. Write user defined Interrupt Service Routines ISR's. Write Terminate but Stay Resident or TSR programs.

5. Control the interrupts by writing programs that control the Programmable Interrupt Controller or PIC.

6. Study the BIOS firmware of the computer system by examining the Power On Self Test or POST, the BIOS Drivers, and the CMOS memory.

7. Write device drivers and install the device drivers.

8. Interface with the mouse and printer. Study the Printer Driver, serial port interfaces, and serial mouse interface.

9. Interface to various ISA Bus devices. Program the DMA controller chip, the Timers, Real Time Clock and configuration RAM.

10. Write programs that will interface the IEEE-488 General Purpose Bus Interface or GPIB with the computer and peripheral electronic devices such as: the digital voltmeter, frequency generator, programmable power supply chips and D/A converter chips.

11. Program the Digital Signal Processor or DSP of the Sound Blaster mixer chip at the hardware level. Use the custom driver for the Sound Blaster provided in the text.

12. Use data compression techniques of coding and decoding and variable code length (Huffman) and adaptive code tables (Lempel-Ziv-Welch or LZW). Use the Microsoft Real Time Compression Interface or MRCI which is a variant of the LZW data compression technique.

13. Program the video hardware, classify the memory models and run the tested and optimized examples of the basic techniques from EGA to TrueColor Adapters. Program the SVGA to maximum effect.

14. Create your own optimized custom graphics library or GRAPH.LIB.

15. System program in all x86 modes: real , protected , and vitual-86. Run a demo program which enters the protected mode kernel and allocates a memory block using the DOS Protected Mode Interface or DPMI.

16. A DPMI host which implements DPMI services is called a server. Write programs that virtualize memory and devices according to a supervisor-user protection model.

17. Describe the Pentium processor architecture, its new features, instructions, registers, and how its internals work. Discuss Pentium-aware optimization using the two pipelines, and the aspects of floating point optimization. Bench test different versions of the supplied programs on disk.

## B. Learning Outcomes with Potential Elements of the Performance:

1. Demonstrate proficiency at writing programs in Assembly Language, utilizing MASM or Turbo Assembler/Debugger, and utilizing assembler macros and macro libraries.

**Potential Elements of the Performance:**

- Edit, assemble, link, and debug using MASM or Turbo debug
- Use Procedure when writing programs
- Pass parameters using the stack for Procedures
- Write macros and add macros to the macro library
- Write Procedures to perform string manipulation

2.

Write program modules that interface with High level languages such as "C". The assembly modules will perform arithmetic processing, table searching, and/or sorting. The "C" level language module will perform Input/ Output (Input from the keyboard ; Output to the screen or monitor).

**Potential Elements of the Performance:**

- write main module in the high level language 'C' and call the assembly sub procedures from the 'C' program
- assemble and link separately compiled OBJECT modules
- employ standard techniques for passing parameters between modules
- write programs that demonstrate calling and passing parameters between a high level language and assembly language written modules
- utilize a librarian to maintain separately assembled modules and utilize the linker to link with library

3.

Interface to the numeric co-processor.

**Potential Elements of the Performance**:

- Write programs that have floating point instruction set requirements.
- Use the REAL data types of short, long, and temporary real

4.

Write user defined Interrupt Service Routines ISR's. Write Terminate but Stay Resident or TSR programs.

**Potential Elements of the Performance**

- Write programs that utilize DOS and BIOS services for performing disk I/O.
- Write programs that utilize DOS and BIOS for controlling the printer.
- Calculate the vector address for an interrupt service routine
- Discuss what happens when the interrupt or INT instruction executes

5.

Control the interrupts by writing programs that control the Programmable Interrupt Controller or PIC.

**Potential Elements of the Performance**

- Write program that continually poll interrupt request lines IR0 to IR7
- switch between slave/ and no slave settings for the ICW3 word, cascade or single mode, level triggered and edge triggered modes for the ICW4 word.
- set the interrupt vector address for the ICW2 word
- set the priority level to be acted upon and the specific rotation (set priority command) for the OCW2 word
- read the Interrupt Request register on the next pulse and poll, no action 0 using the OCW3 word
- disable and enable devices using the OCW1 word

6.

Study the BIOS firmware of the computer system by examining the Power On Self Test or POST, the BIOS Drivers, and the CMOS memory.

**Potential Elements of the Performance**

- perform the POST test and document settings
- examine the BIOS Drivers installed
- change the CMOS settings

7.

Write device drivers and install the device drivers.

**Potential Elements of the Performance**

- List the device driver fundamentals
- install device drivers (both character and block drivers)
- describe the structure of a device driver: driver header, strategy program, interrupt program
- list the device driver functions
- examine the resident drivers and the affect on the allocation of DOS device drivers in RAM memory as you install another device driver

8.

Interface with the mouse and printer. Study the Printer Driver, serial port interfaces, and the serial mouse interface.

**Potential Elements of the Performance**

- Describe parallel interfaces
- Describe serial interfaces
- Use the Int 5, 10h, 0fh, and 17h to control the printer using BIOS service routines
- Use the Int 21 function calls 3dh, 40h, 3eh, and 44h to control the printer using DOS service routines
- Use the mouse co-ordinate Interface functions of Int 33h
- Run the simple screen pencil program provided in the text
- Use the BIOS interrupt 14h serial Port I/O and the mouse driver imitation program provided

9.

Interface to various ISA Bus devices. Program the DMA controller chip, the Timers, Real Time Clock and configuration RAM.

**Potential Elements of the Performance**

- Study how the interrupt controller works
- write interrupt programs that set the interrupt controller operating modes
- write programs that set the Direct Memory Access or DMA controller
- write programs that set the Timer chip using modes timer 0, timer 1, and timer 2
- use the real time clock and configuration RAM

10.

Write programs that will interface the IEEE-488 General Purpose Bus Interface or GPIB with the computer and peripheral electronic devices such as: the digital voltmeter, frequency generator, programmable power supply chips and D/A converter chips.

**Potential Elements of the Performance**

- Write programs that will interface the GPIB with other computers and peripheral electronic devices such as: the digital voltmeter, frequency generator, programmable power supply chips and D/A converter chips.

11.

Program the Digital Signal Processor or DSP of the Sound Blaster mixer chip at the hardware level. Use the custom driver for the Sound Blaster provided in the text.

**Potential Elements of the Performance**

- Describe the principles of sound generation theory: the straight forward approach, the subtractive method, FM synthesis, wave form synthesis, and sampling
- Describe the structure of a typical sound card
- use standard drivers to work with digitized sound in memory
- play and record WAV files
- play MIDI music and VOC Voices together
- customize sounds by writing your own driver for DSP programming

12.

Use data compression techniques of coding and decoding and variable code length (Huffman) and adaptive code tables (Lempel-Ziv-Welch or LZW). Use the Microsoft Real Time Compression Interface or MRCI which is a variant of the LZW data compression technique.

**Potential Elements of the Performance**

- Describe basic data compression techniques
- perform decompression using the Huffman algorithm, and the LZW compression algorithm
- perform the arithmetic compression using the provided program in the text

13.

Program the video hardware, classify the memory models and run the tested and optimized examples of the basic techniques from EGA to TrueColor Adapters. Program the SVGA to maximum effect.

**Potential Elements of the Performance**

- program SVGA's using the industry standard Video Electronics Standards Association or VESA support
- program SVGA's without the VESA support
- perform advance video programming working with the CRT controller and the VGA register set

14.

Create your own optimized custom graphics library or GRAPH.LIB.

**Potential Elements of the Performance**

- determine graphics library design compatibility
- explore GRAPH.LIB routines for mode switching, line drawing, clipping, and filling polygons
- examine PC animation techniques: sprite and other practical animation techniques

15.

System program in all x86 modes: real , protected , and vitual-86. Run a demo program which enters the protected mode kernel and allocates a memory block using the DOS Protected Mode Interface or DPMI.

**Potential Elements of the Performance**

- The application programmer Interface or API is used for mode switching, local descriptor table management, memory management (conventional and extended memory), controlling interrupt system, accessing CPU control registers, and co-ordination with real mode applications.
- Run the demo program provided in the text

16.

A DPMI host which implements DPMI services is called a server. Write programs that virtualize memory and devices according to a supervisor-user protection model.

**Potential Elements of the Performance**

- Overview the existing protocols for memory above 1Mb and memory below 1 Mb
- Program using the expanded memory manager API ( Application Programmer Interface)
- Program using the eXtended Memory Standard or XMS
- Program in protected mode for multitasking
- Overview the DOS Protected Mode Interface or DPMI services

17.

Describe the Pentium processor architecture, its new features, instructions, registers, and how its internals work. Discuss Pentium-aware optimization using the two pipelines, and the aspects of floating point optimization. Bench test different versions of the supplied programs on disk.

**Potential Elements of the Performance**

- Describe the Pentium architecture as to structural scheme of the chip and the code and data caches
- Discuss the U and V pipelines and the pipelined floating point unit or FPU
- bench test the Pentium-aware optimization for integer and floating point optimization
- Discuss the vitual-86 mode

## 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 order below. The text book chapters are also included.

1. Review of assembly language chapter 1
2. Programming the Intel 80386/80486 Processors chapter 2
3. Fundamental System programming techniques (PIC. TSR, BIOS and CMOS) chapter 3
4. Writing Device Drivers chapter 5
5. Interfacing to Other Devices ( printers, serial and parallel, BIOS and DOS techniques) serial mouse interface chapter 6
6. Interfacing to the ISA Bus devices ( DMA controller, Timer, PIC) chapter 7
7. Programming Sound ( WAV and VOC files) Digital Signal programming or DSP programming chapter 8
8. Data Compression Techniques ( Huffman, LZW, Arithmetic compression) chapter 10
9. PC Video Architecture ( VESA and advanced video programming) chapter 11
10. Advanced Video techniques ( the Graphics library routines) chapter 12
11. Memory above and below 1 Mb chapter 13 and 14
12. Programming in Protected Mode ( real, protected and virtual -86 modes) chapter 15 and 16
13. Pentium Programming chapter 18
14. Assembly language libraries chapter 19
15. MicroSoft real time Compression Interface or MRCI appendix A
16. Graphics FILE Formats (bmp, pcx, targa, gif, jpeg) Appendix B
17. DOS Protected Mode Interface or DMPI Function Reference Appendix C
18. Sound File Formats ( VOC and WAV) Appendix D
19. GPIB bus interface standard IEEE -488

## IV. REQUIRED STUDENT RESOURCES:

1. TEXTBOOK: **"Master Class Assembly Language"**
by WROX Press Ltd
ISBN 1 - 874416-34-6

2. DISKETTES: five 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

## V. METHOD OF EVALUATION:

| | |
|---|---|
| 2 THEORY TESTS (30% each) | 60% |
| ASSIGNMENTS | 25% |
| QUIZZES AND PRACTICAL TESTS | 15% |

(The percentages shown above may have to be adjusted to accurately evaluate student skills. Students will be notified of any changes made.)

### TESTS

Written tests will be announced about one week in advance. Quizzes may be conducted without advance warning. No "re-write" opportunities exist for quizzes not written.

## 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 mandatory** 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.

A penalty for late assignments will be applied unless there are extenuating circumstances. A 10% per week penalty will be applied. After 4 weeks late assignments will not be accepted for credit.

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. Where plagiarism or copying is found and it is impossible to determine whose original work it is, a mark of zero will be assigned to all assignments involved.

## ATTENDANCE:

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

The following letter grades will be assigned in accordance with the School of Business and Hospitality 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 |
| X | | a temporary grade |

An 'X' grade is limited in use to rare 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 submission 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.

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 students attendance has been satisfactory.
2. An overall average of at least 45% 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. 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 is deemed 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.

6. Students must achieve a passing grade in **both** the assignment (25%) and the test (60%) portions of the course.

## VII. PRIOR LEARNING ASSESSMENT:

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