SAULT COLLEGE OF APPLIED ARTS & TECHNOLOGY
SAULT STE. MARIE, ONTARIO

COURSE OUTLINE

COURSE TITLE: ADVANCED APPLICATIONS PROGRAMMING

CODE NO.: EDP229-6 SEMESTER: FOUR

PROGRAM: COMPUTER PROGRAMMER

AUTHOR: DENNIS OCHOSKI

DATE: SEPTEMBER 1995

PREVIOUS OUTLINE DATED: JANUARY 1995

New: ________ Revision: x

APPROVED: ______________________
DEAN, SCHOOL OF BUSINESS & HOSPITALITY

95-08-21
DATE

ADVANCED PROGRAMMING LANGUAGES — EDP229

COURSE NAME — COURSE CODE

Length of Course: 5 periods per week for one semester

Required Resources:

Text: *C Programming Using Turbo C++*, by Robert Lafore

Disks: 2, 3 1/2" floppy diskettes

Philosophy/Goals:

This course will provide students with an opportunity to develop their programming skills using a "leading-edge" language, C. C is expected to emerge as the dominant programming language of the mid-to-late 1990s. The course will re-emphasize the use of structured programming techniques and proper program design.

Special Notes:

1. Students are advised to maintain a copy of all files on a backup disk. Loss of an assignment due to a lost or damaged disk is not an acceptable reason for a late or incomplete assignment.

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

3. 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.

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

ADVANCED PROGRAMMING LANGUAGES	EDP229
COURSE NAME	COURSE CODE

Student Evaluation:

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

Tests:

Test #1 - Modules 1 & 2	10%	
Test #2 - Module 3	15%	
Test #3 - Module 4	20%	
Test #4 - Modules 5,6 & 7	20%	
		65%

Assignments:

Asgn #1 - Modules 1 & 2	5%	
Asgn #2 - Module 3	5%	
Asgn #3 & #4 - Module 4	10%	
Asgn #5 - Module 5	10%	
Asgn #6 - Module 6	10%	
Asgn #7 - Module 7	5%	
		35%
		100%

Grading:

A+	90 - 100%
A	80 - 89%
B	70 - 79%
C	60 - 69%
R	UNDER 60% - Repeat Course

Course Objectives:

1. Build upon programming skills acquired in previous semesters.

2. Learn a systems implementation language of choice in industry used for both systems and applications programming.

3. Further develop structured programming techniques and design.

4. Develop solutions to business information problems using C.

ADVANCED PROGRAMMING LANGUAGES	EDP229
COURSE NAME	COURSE CODE

Module 1: Introduction to C Programming (chapters 1 and 2)

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

1. understand the C program development environment (IDE - Integrated Development Environment).

2. become familiar with the structure of C programs.

3. become familiar with fundamental data/variable types.

4. become familiar with arithmetic and relational operators.

5. understand the use of input/output functions such as scanf(), getche(), and printf().

Module 2: Control Structures (chapters 3 and 4)

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

1. use the *while*, *do/while*, and *for* looping structures to execute statements in a program repeatedly.

2. use the *break* and *continue* program control statements.

3. use the *if*, *if/else*, and *switch* selection structures to choose among alternative actions.

4. use the increment, decrement, assignment, and logical operators.

Module 3: Functions
(chapter 5)

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

1. understand how to construct programs modularly from small pieces called functions.
2. become familiar with the common math functions available in the C standard library.
3. write and use custom-designed functions.
4. understand the mechanisms used to pass information between functions.
5. apply simulation techniques using random number generation.

Module 4: Arrays, Strings, and Pointers
(chapters 6 and 7)

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

1. understand the array data structure.
2. understand the use of arrays to store, sort, and search lists and tables of values.
3. pass arrays to functions.
4. understand basic sorting techniques.
5. declare and use multiple-subscript arrays.
6. use pointers in various capacities.
7. understand the class relationship among pointers, arrays, and strings.
8. declare and use arrays of strings.

Module 5: Keyboard and Cursor Control
(chapter 8)

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

1. become familiar with C's extended keyboard codes.

2. understand and use Turbo C++ text-window functions.

3. use command-line arguments.

4. perform redirection of input and output.

Module 6: Structures and Files
(chapters 9 and 11)

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

1. create and use the various types of structures such as, simple structures, nested structures, array of structures, and pointer structures.

2. use the functions of the string handling library (string.h).

3. create, read from, write to, and update files.

4. become familiar with sequential access files.

5. perform input from, and output to, character string arrays.

ADVANCED PROGRAMMING LANGUAGES	EDP229
COURSE NAME	COURSE CODE

Module 7: Introduction to Turbo C++ Graphics (chapter 10)

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

1. initialize the graphics system.
2. create screen displays of rectangles, ellipses, and polygons.
3. incorporate image filling and patterns into graphics images.
4. use bit imaging and animation techniques.
5. add text to graphics images.