SAULT COLLEGE OF APPLIED ARTS AND TECHNOLOGY
SAULT STE. MARIE, ON

COURSE OUTLINE

COURSE TITLE: COMPUTER PROGRAMMING 2

CODE NO.:CSD101 SEMESTER:WINTER 96

PROGRAM: COMPUTER ENGINEERING TECHNOLOGY
COMPUTER PROGRAMMER

AUTHOR: GERRY DAVIES/DENNIS OCHOSKI

DATE:NOVEMBER 1995_____ PREVIOUS OUTLINE DATED:_____

APPROVED: _____ 96-01-02
DEAN DATE

CSD 101 - COMPUTER PROGRAMMING 2

TOTAL CREDITS 4
PREREQUISITE(S):NONE

I. PHILOSOPHY/GOALS: This course is intended to extend the foundation of computer programming skills needed in the computer studies area. It is the second course in the C language, and develops the students problem-solving, computer programming, and software utilization skills.

II. STUDENT PERFORMANCE OBJECTIVES (OUTCOMES):

Upon successful completion of this course the student will be able to:

1. Discuss the concepts of complex data, including scope and class of data, arrays, strings, structures, union, data files and the use of pointers.
2. Develop algorithms to solve problems using complex data, macros, and user-written functions, and describe those algorithms using pseudocode.
3. Discuss the use of include files, object libraries, and the linker, and be able to create programs using user-written, separately-compiled functions.
4. Write programs using the concepts and programming techniques covered in chapters 6 through 13 of the text at a level of complexity similar to the assigned problems.

III. TOPICS TO BE COVERED:

1. Functions and arrays in C.
2. Using pointers and strings.
3. The use of complex data and files.

CSD 101 - COMPUTER PROGRAMMING 2

IV. LEARNING ACTIVITIES

BLOCK 1 Functions and Arrays in C

At the end of this block, the student will be able to:

1. Define or describe the meaning of the following terms:
   Global variable, local variable, auto class, static class, external class, register class, pass by value, pass by reference, single-dimensional array, subscript, index value, null character, two-dimensional array.
2. Discuss the concept of scope and class in C, and the appropriate use of the various classes.
3. Discuss the use of some of the standard library functions supplied with Turbo C++ such as the math, string and Ctype libraries, and how to determine the libraries that are available and which library a particular function is in.
4. Discuss the concepts of passing arguments to functions by value and by reference.
5. Write programs that use user-written functions, include files, and object libraries.
6. Discuss the concepts of one and two dimensional arrays, and describe how they are declared, initialized and passed to functions.
7. Write, test, and debug programs using the concepts and techniques of chapters six and seven at a level similar to the assigned problems.

BLOCK 2 Pointers and Strings

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

1. Define or describe the meaning of the following terms:
   offset, pointer constant, null character.
2. Discuss the use of pointers and pointer arithmetic, and be able to use pointers with arrays.
3. Discuss the concept of strings in C, and how they are implemented.
4. Discuss the use of the following string functions: gets, puts, strcopy, strcat, strchr, strcmp and strlen.
5. Write, test and debug programs using the concepts and techniques of chapters 8 and 9 at a level similar to the assigned problems.

BLOCK 3 Data Structures and Files

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

1. Define or describe the meaning of the following terms: structure, member, content, populating the structure, record, union, open, close, append, internal pointer, stdin, stdout, stderr.
2. Discuss the concepts of structures in C, and describe how they are used in programs, including the use of arrays of structures, and the methods of passing and returning structures from functions.
3. Discuss basic file concepts, including the concepts of opening and closing files, and reading, writing, appending and updating data in files.
4. Describe the use of the following C functions: fopen, fclose, fputc, fputs, fprintf, fgetc, fgets, fscanf.
5. Discuss the concepts of standard device files.
6. Write, test, and debug programs utilizing the concepts and techniques of chapters 10 and 11 at a level similar to the assigned problems.

V. EVALUATION METHODS:

The mark for this course will be arrived at as follows:

Tests	**45%**
Quizzes	**25%**
Assignments and practical work	**30%**
total	**100%**

Assignments must be submitted by the due date according to the specifications of the instructor. Late assignments will normally be given a mark of zero. Late assignments will only be marked at the discretion of the instructor in cases where there were extenuating circumstances. Lab work must be completed to the satisfaction of the instructor to successfully complete the course.

The instructor reserves the right to modify the assessment process to meet any changing needs of the class. Consultation with the class will be done prior to any changes.

The grading scheme used will be as follows:

A+	90 - 100%	Outstanding achievement
A	80 - 89%	Excellent achievement
B	70 - 79%	Average achievement
C	56 - 69%	Satisfactory achievement
R	Repeat	
X	Incomplete.	

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 Dean, and reverts to an R grade if not upgraded within a specified time.

The method of upgrading an incomplete grade is at the discretion of the instructor, and may consist of such things as make-up work, rewriting tests, and comprehensive examinations.

VI. PRIOR LEARNING ASSESSMENT:

Students who wish to apply for advanced credit in the course should consult the instructor.

VII. REQUIRED STUDENT RESOURCES

Text: A First Book of ANSI C
Fundamentals of C Programming
Gary Bronson and Stephen Menconi
West Publishing

VIII. SPECIAL NOTES

Students with special needs (eg. physical limitations, visual impairments, hearing impairments, learning disabilities) are encouraged to discuss required accommodations confidentially with the instructor.

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