

SAULT COLLEGE OF APPLIED ARTS AND TECHNOLOGY

SAULT STE. MARIE, ONTARIO



Sault College

COURSE OUTLINE

COURSE TITLE: COMPUTER PROGRAMMING 2

CODE NO. : CSD1010 **SEMESTER:** 2

PROGRAM: ALL COMPUTER STUDIES PROGRAMS

AUTHOR: Dennis Ochoski

DATE: January, 2003 **PREVIOUS OUTLINE DATED:** Jan. 2002

APPROVED:

	_____	_____
	DEAN	DATE
TOTAL CREDITS:	<u>4</u>	
PREREQUISITE:	<u>CSD1000</u>	
HOURS/WEEK:	<u>4</u>	

Copyright ©2000 The Sault College of Applied Arts & Technology
Reproduction of this document by any means, in whole or in part, without prior written permission of Sault College of Applied Arts & Technology is prohibited.
For additional information, please contact Kitty DeRosario,
School of Trades & Technology
(705) 759-2554, Ext. 642

COURSE NAME

COURSE CODE

I. COURSE DESCRIPTION: 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/C++ programming language, and further develops the student's problem-solving, computer programming, and software utilization skills.

II. TOPICS TO BE COVERED:

1. Advanced Concepts with User-defined Functions.
2. Arrays/Tables.
3. Pointers.
4. Advanced Concepts with Characters and Strings.
5. Data Structures.
6. Files.
7. Other Concepts: input and output with *scanf()* and *printf()*
Bitwise operators

III. LEARNING OUTCOMES AND ELEMENTS OF THE PERFORMANCE:

Upon successful completion of this course the student will demonstrate the ability to:

1. Discuss and create user-written, independently-compiled functions that pass and receive values.

(Gaddis: chapter 6)

This learning outcome will comprise approximately **33%** of the course.

Elements of the performance:

- define and apply the concepts of the following terms:

scope	calling vs called functions	function prototypes
local vs global variables	pass by value	return statement
class	pass by reference	overloaded functions
auto vs static variables	arguments/parameters	

- develop modularized, structured programs by creating user-written functions
- discuss and apply the concepts of ‘passing’ arguments to called functions by value
- discuss and apply the concept of ‘returning’ values to calling functions
- discuss and apply the concepts of ‘passing’ arguments to called functions by reference
- write, test, and debug programs containing functions

2. Develop algorithms and write C programs to solve problems involving tables/arrays.

(Gaddis: chapters 7 and 8)

This learning outcome will comprise approximately **20%** of the course.

Elements of the performance:

- define and apply the concepts of the following terms:

one-dimensional array	index value	subscript
two-dimensional array	null character	

COURSE NAME

COURSE CODE

Elements of the performance(cont'd):

- discuss the purpose and concepts relating to one- and two-dimensional arrays
- declare and initialize both numeric and character arrays
- access and process array elements
- pass arrays between functions
- write, test, and debug programs containing arrays

3. Develop algorithms to solve problems involving the use of pointers.
(Gaddis: chapter 9)

This learning outcome will comprise approximately **20%** of the course.

Elements of the performance:

- discuss and apply the concept of pointers and pointer arithmetic
- apply the concept of pointers to arrays
- discuss and apply the concept of pointers in C/C++
- discuss and apply the concept of ‘passing’ arguments to called functions by address
- write, test, and debug programs using pointers

4. Discuss and apply the concepts of character and string manipulation with reference to C/C++ library functions.
(Gaddis: chapter 10)

This learning outcome will comprise approximately **10%** of the course.

Elements of the performance:

- discuss and apply character-based I/O functions such as:

cin.get() tolower() toupper()

- discuss and apply character-testing functions such as:

isalpha() isalnum() isdigit()
islower() isupper()

COURSE NAME

COURSE CODE

Elements of the performance(cont'd):

- discuss and apply string functions such as:

strcat() strcmp() strlen() strcpy()

- discuss and apply string/numeric conversion functions such as:

atoi() atof() atol() itoa()

- write, test, and debug programs containing character and string functions

5. Develop algorithms to solve problems involving the use of data structures.
(Gaddis: chapter 11)

This learning outcome will comprise approximately **8%** of the course.

Elements of the performance:

- define and apply the concepts of the following terms:

structure member record internal pointer

- discuss the concept of structures in C/C++
- declare and initialise a structure
- access and process structure members
- apply the use of arrays of structures
- discuss and apply methods of passing and returning structures to and from functions
- write, test, and debug programs containing structures

6. Develop algorithms to solve problems involving the use of file manipulation.

COURSE NAME

COURSE CODE

(Gaddis: chapter 12)

This learning outcome will comprise approximately **7%** of the course.

Elements of the performance:

- define and apply the concepts of the following terms:

file open close append

- create a disk file
- write data to, and, read data from a disk file
- perform disk I/O with records
- understand, create, and manipulate sequential files
- write, test, and debug programs containing files

7. Discuss and apply other concepts such as input/output using *scanf()/printf()*, and, bitwise operators used to manipulate data.
(Appendix D and lecture notes)

This learning outcome will comprise approximately **2%** of the course.

Elements of the performance:

- apply the input/output functions *scanf()/printf()* in place of *cin/cout*
- define and apply the concepts of the following terms:

TRUE bitwise OR bit shifting
FALSE bitwise XOR bitwise complement
bit manipulation bitwise AND

- discuss the concept of truth tables
- apply bitwise operators

IV. EVALUATION METHODS:

COURSE NAME

COURSE CODE

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

Quizzes:

outcome #1	23%
outcome #2	15%
outcomes #3, #4	20%
outcomes #5, #6, #7	12%

Assignments:

outcome #1	10%
outcome #2	5%
outcomes #3, #4	10%
outcomes #5, #6, #7	<u>5%</u>

Total	100%
-------	------

The grading scheme used will be as follows:

A+	90 - 100%	Outstanding achievement
A	80 - 89%	Excellent achievement
B	70 - 79%	Average achievement
C	60 - 69%	Satisfactory achievement
R	Repeat	
X	Incomplete	A temporary grade limited to special circumstances that have prevented the student from completing objectives by the end of the semester. An X grade reverts to an R grade if not upgraded within a specified time.

COURSE NAME

COURSE CODE

ELIGIBILITY FOR XGRADES/UPGRADING OF INCOMPLETES

When a student's course work is incomplete or final grade is below 60%, 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 50% 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: **The opportunity for an X grade is usually reserved for those with extenuating circumstances.** The nature of the upgrading requirements will be determined by the instructor and may involve one or more of the following: completion of existing labs and assignments, completion of additional assignments, re-testing on individual parts of the course or a comprehensive test on the entire course.

ASSIGNMENTS

Required format for lab assignments will be detailed by the instructor before labs are assigned.

ATTENDANCE:

Absenteeism will affect a student's ability to succeed in this course. Absences due to medical or other unavoidable circumstances should be discussed with the instructor. There will be an attendance factor included in the lab evaluation.

V. SPECIAL NOTES

1. In order to pass this course the student must obtain an overall quiz average of **60%** or better, as well as, an overall assignment average of **60%** or better. A student who is not present to write a particular quiz, and does not notify the instructor beforehand of their intended absence, may be subject to a zero grade on that quiz.
2. 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.
3. Any assignment submissions deemed to be copied will result in a **zero** grade being assigned to

COURSE NAME

COURSE CODE

all students involved in that particular incident.

4. 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.
5. Students with special needs (eg. physical limitations, visual impairments, hearing impairments, learning disabilities) are encouraged to discuss required accommodations confidentially with the instructor and/or the Special Needs office. Visit Room E1204 or call extension 493, 717, or 491 so that support services can be arranged for you.
6. Your instructor reserves the right to modify the course as he/she deems necessary to meet the needs of students.

VI. PRIOR LEARNING ASSESSMENT:

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

VII. REQUIRED RESOURCES/TEXTS/MATERIALS

Text: Brief Version of Starting Out With C++, 3rd edition
 by Tony Gaddis

Diskettes: minimum of 3, 3 1/2"

V. SPECIAL NOTES

1. In order to pass this course the student must obtain an overall **quiz** average of 60% or better, as well as, an overall **assignment** average of 60%.
2. Assignments must be submitted by the specified due date. Late assignments will be penalized at a 10% reduction per day, up to 3 days late, after which the assignment will be given a mark of zero. Assignments submitted more than 3 days late may be marked at the discretion of the instructor in cases where there are extenuating circumstances.
3. 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.
4. Their will be **no** re-write of any quiz unless the instructor feels there are extenuating circumstances.