SAULT COLLEGE OF APPLIED ARTS AND TECHNOLOGY

SAULT STE. MARIE, ONTARIO

Sault College

COURSE OUTLINE

COURSE TITLE:	**COMPUTER PROGRAMMING 2**
CODE NO. :	**CSD101** **SEMESTER: 2**
PROGRAM:	**ALL COMPUTER STUDIES PROGRAMS**
AUTHOR:	**Dennis Ochoski**
DATE:	**Jan, 2007** **PREVIOUS OUTLINE DATED: Jan, 2006**
APPROVED:	
	DEAN **DATE**
TOTAL CREDITS:	**4**
PREREQUISITE(S):	**CSD100**
HOURS/WEEK:	**4**

COMPUTER PROGRAMMING 2	CSD101
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. The Debugger.
3. Arrays/Tables.
4. Pointers.
5. Advanced Concepts with Characters and Strings.
6. Data Structures.
7. Files.
8. 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. (Johnson: chapters 4, 5 and 8)

   This learning outcome will comprise approximately **38%** of the course.

   Elements of the performance:

   - discuss and apply the concept of pointers and pointer arithmetic
   - discuss and apply the concept of pointers in C/C++
   - 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. Debug program logic errors using the C++ Debugger. (Johnson: Appendix J)

   This learning outcome will comprise approximately **5%** of the course.

   Elements of the performance:

   - execute code one line at a time using the Step Debugger
   - use the following stepping options: **Go, Step Into, Step Over, Step Out, Watch,** and **Run to Cursor**
   - define, as well as, insert and remove breakpoints

3. Develop algorithms and write C++ programs to solve problems involving tables/arrays. (Johnson: chapter 6)

   This learning outcome will comprise approximately **15%** 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	

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

4. Discuss and apply the concepts of character and string manipulation with reference to C/C++ library functions. (Johnson: chapter 6 and Appendix G)

   This learning outcome will comprise approximately **6%** of the course.

   Elements of the performance:

   - discuss and apply character-based functions such as:

cin.get( )	tolower( )	toupper( )	isalpha( )
isdigit( )	isalnum( )	islower( )	isupper( )

   - discuss and apply string functions such as:

strcat( )	strcmp( )	strlen( )	strcpy( )
atoi( )	atof( )	atol( )	itoa( )

   - understand and utilize the C++ string class and its associated functions to declare string variables and manipulate string values
   - write, test, and debug programs containing character and string functions

5. Develop algorithms to solve problems involving the use of data structures. (Johnson: chapter 7)

   This learning outcome will comprise approximately **11%** 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
   - 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. (Johnson: Appendix F)

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

   Elements of the performance:

   - define and apply the concepts of the following terms:

     file open read close write append

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

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

This learning outcome will comprise approximately **5%** of the course.

Elements of the performance:

- apply the input/output functions *scanf()/printf()* in place of *cin/cout*
- discuss the concept of truth tables
- apply bitwise operators
- define and apply the concepts of the following terms:

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

IV. REQUIRED RESOURCES/TEXTS/MATERIALS

Text: C++ Programming Today
by Barbara Johnston
ISBN: 1-13-085375-5

V. EVALUATION PROCESS/GRADING SYSTEM:

The following semester grades will be assigned to students in postsecondary courses:

Topic	Quizzes	Assignments	Total
Debugger, user-defined functions	15%	10%	25%
Arrays. pointers	15%	10%	25%
Arrays, strings	15%	10%	25%
Structures, files	15%	10%	25%
	65%	25%	100%

Grade	Definition	Grade *Point Equivalent*
A+	90 - 100%	4.00
A	80 - 89%	4.00
B	70 - 79%	3.00
C	60 - 69%	2.00
D	50 - 59%	1.00
F (Fail)	below 50%	0.00

CR (Credit)	Credit for diploma requirements has been awarded.
S	Satisfactory achievement in field /clinical placement or non-graded subject area.
U	Unsatisfactory achievement in field/clinical placement or non-graded subject area.
X	A temporary grade limited to situations with extenuating circumstances giving a student additional time to complete the requirements for a course.
NR	Grade not reported to Registrar's office.
W	Student has withdrawn from the course without academic penalty.

VI. SPECIAL NOTES:

Special Needs:
If you are a student with special needs (e.g. physical limitations, visual impairments, hearing impairments, or learning disabilities), you are encouraged to discuss required accommodations with your professor and/or the Special Needs office. Visit Room E1101 or call Extension 2493 so that support services can be arranged for you.

Retention of Course Outlines:
It is the responsibility of the student to retain all course outlines for possible future use in acquiring advanced standing at other postsecondary institutions.

Plagiarism:
Students should refer to the definition of “academic dishonesty” in *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/program, as may be decided by the professor/dean. In order to protect students from inadvertent plagiarism, to protect the copyright of the material referenced, and to credit the author of the material, it is the policy of the department to employ a documentation format for referencing source material.

Course Outline Amendments:
The professor reserves the right to change the information contained in this course outline depending on the needs of the learner and the availability of resources.

Other Pertinent Information

1. In order to pass this course the student must obtain an overall quiz average of **50%** or better, as well as, an overall assignment/project average of **50%** or better. A student who is not present to write a particular quiz, and does not notify the professor beforehand of their intended absence, may be subject to a zero grade on that quiz.

2. There will be **no** supplemental or make-up quizzes/tests in this course.

3. Assignments must be submitted by the due date according to the specifications of the professor. Late assignments will normally be given a mark of zero. Late assignments will only be marked at the discretion of the professor in cases where there were extenuating circumstances.

VI. SPECIAL NOTES: (cont'd)

4. Any assignment/projects submissions, deemed to be copied, will result in a **zero** grade being assigned to **all** students involved in that particular incident.

5. It is the responsibility of the student to ask the professor to clarify any assignment requirements.

6. The professor reserves the right to modify the assessment process to meet any changing needs of the class.

VII. PRIOR LEARNING ASSESSMENT:

Students who wish to apply for advanced credit in the course should consult the professor. Credit for prior learning will be given upon successful completion of a challenge exam or portfolio.

VIII. DIRECT CREDIT TRANSFERS:

Students who wish to apply for direct credit transfer (advanced standing) should obtain a direct credit transfer form from the Dean's secretary. Students will be required to provide a transcript and course outline related to the course in question.