SAULT COLLEGE OF APPLIED ARTS AND TECHNOLOGY

SAULT STE. MARIE, ONTARIO

Sault College

COURSE OUTLINE

COURSE TITLE: **Computer Programming II**

CODE NO. : **CSD101**   SEMESTER: **2**

PROGRAM: **All Computer Studies Programs**

AUTHOR: **Tycho Black, Dennis Ochoski, Bazlur Rasheed**

DATE: **Jan, 2001**   PREVIOUS OUTLINE DATED: **Jan, 2000**

APPROVED:

_____   _____
                                        DEAN   DATE

TOTAL CREDITS: **5**

PREREQUISITE(S): **CSD100**

HOURS/WEEK: **4**

Computer Programming II   **CSD101**
Course Name   Code No.

I.   COURSE DESCRIPTION:

In this second course in Computer Programming, students continue their
development of skills in program modularization, design and debugging,
expanding their C/C++ language skills to include the use of functions, arrays and
pointers, file handling, structures and libraries. Problem-solving skills are
developed through programming assignments of increasing complexity.

II. A   LEARNING OUTCOMES:

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

1.   Apply bitwise operators for low-level data-manipulation.

2.   Write programs applying the concepts of Pointers and References.

3.   Write functions demonstrating function call by value, reference and address
      and utilizing various storage classes - auto/static, register.

4.   Apply concepts of Arrays/Tables in solving problems.

5.   Use pointers and strings with C-style string manipulation library functions.

6.   Group data using structures and differentiate between structure and class.

7.   Write programs applying the concepts of File handling.

8.   Apply the concepts of memory allocation in C/C++.

9.   Use preprocessor directives appropriately.

10   Write programs which can take command line arguments.

11   Use the make utility to manage large programs.

Computer Programming II   **CSD101**
Course Name   Code No.

II. B   LEARNING OUTCOMES AND ELEMENTS OF THE PERFORMANCE:

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

1.   Apply bitwise operators for low-level data-manipulation.

     Potential Elements of the Performance:

     - define and apply the concepts of the following terms:
             bitwise OR   bit shifting
             bitwise XOR   bitwise complement
             bitwise AND   Mask
     - write, test, and debug programs using the above operators

2.   Write programs applying the concepts of Pointers and References.

     Potential Elements of the Performance:
     - define and apply the concepts of the following terms:
             Pointers (address variables)   address operator
                                             dereferencing operator
             Reference variable   reference operator
             Pointer arithmetic   Null pointer

3.   Write functions demonstrating function call by value, reference and address and utilizing various storage classes - auto/static, register.

     Potential Elements of the Performance:

     - define and apply the concepts of the following terms:
             calling vs. called functions call by value
             call by address   call by reference
             auto vs. static variables   arguments/parameters
         - 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 using the above operators

<u>Computer Programming II</u> <u>**CSD101**</u>
Course Name Code No.

4. Apply concepts of Arrays/Tables in solving problems.

   <u>Potential Elements of the Performance</u>:

   - 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 initialise both numeric and character arrays
     - pass arrays between functions
   - write, test, and debug programs containing arrays

5. Use pointers and strings with C-style string manipulation library functions.

   <u>Potential Elements of the Performance</u>:

   - discuss and apply the concept of pointers to arrays and strings
     - discuss and apply the concept of strings and pointers in C/C++
     - discuss and apply the use of the following string functions: strcpy,
       strcat, strcmp
   - write, test, and debug programs using pointers and strings

6. Group data using structures and differentiate between structure and class.

   <u>Potential Elements of the Performance</u>:
   - define and apply the concepts of the following terms:

     structure   record   append
     member   open   internal pointer
     record   close   class

     - discuss the concept of structures in C/C++
     - 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 and files

7. Write programs applying the concepts of File handling.

   - create a disk file
     - write data to, and, read data from a disk file
     - perform disk I/O with records

Computer Programming II   **CSD101**
Course Name   Code No.

- discuss and apply the use of the following functions: stdin, stdout, and stderr
- understand, create, and manipulate sequential files
- write, test, and debug programs containing structures and files

8. Apply the concepts of memory allocation in C/C++.

- define and apply the concepts of the following functions and operators:
  malloc(), realloc(), new and delete operators

9. Use preprocessor directives appropriately.

- define and apply the concepts of the following preprocessor directives:
  #define, #undef, #if, #endif, Conditional compilation
  Macros

10 Write programs which can take command line arguments.

- define and apply the concepts of the following terms:
  int main(int argc, char** argv), command line arguments

11 Use the make utility to manage large programs.

- Practice managing large program using make utility

III.   TOPICS:

1. Bitwise operators.

2. Pointers and References.

3. User-defined functions

4. Arrays/Tables.

5. Pointers and strings.

6. Structures and Classes

7. File handling

8. Utilities

IV.   REQUIRED RESOURCES/TEXTS/MATERIALS:

Textbooks:
1.   "Programming and Problem Solving with C++, 2nd Ed "; by- Nell Dale,
        Chip Weems, Mark Headington;
        Jones and Bartlett Publishers, 2000,   ISBN: 0-7637-1523-9
        (package including Visual C++ 6.0 Student Ed and Student Lecture
        Companion)

Other Resources:
        Internet,
        Instructor handouts.

V.   EVALUATION PROCESS/GRADING SYSTEM:

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

3 Written Tests @ 20% each	60%
Lab Assignments, Projects, Lab attendance and Quizzes	<u>40%</u>
Total	100%

ELIGIBILITY FOR X GRADES/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:

The student's attendance has been satisfactory.
An overall average of at least 50% has been achieved.
The student has not had a failing grade in all of the theory tests taken.
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.

Computer Programming II   **CSD101**
Course Name   Code No.

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.

The following semester grades will be assigned to students in post-secondary courses:

Grade	Definition	Grade Point Equivalent
A+	90 - 100%	4.00
A	80 - 89%	3.75
B	70 - 79%	3.00
C	60 - 69%	2.00
R (Repeat)	59% or below	0.00
CR (Credit)	Credit for diploma requirements has been awarded.	
S	Satisfactory achievement in field placement or non-graded subject areas.	
U	Unsatisfactory achievement in field placement or non-graded subject areas.	
X	A temporary grade. This is used in limited situations with extenuating circumstances giving a student additional time to complete the requirements for a course (see *Policies & Procedures Manual – Deferred Grades and Make-up*).	
NR	Grade not reported to Registrar's office. This is used to facilitate transcript preparation when, for extenuating circumstances, it has not been possible for the faculty member to report grades.	

Computer Programming II   **CSD101**
Course Name   Code No.

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 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.

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 post-secondary 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.

Substitute course information is available in the Registrar's office.

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.