SAULT COLLEGE OF APPLIED ARTS AND TECHNOLOGY

SAULT STE. MARIE, ONTARIO

Sault College

COURSE OUTLINE

COURSE TITLE: INTRODUCTION TO OBJECT-ORIENTED PROGRAMMING

CODE NO. : CSD205   **SEMESTER:** 3

PROGRAM: COMPUTER ENGINEERING TECHNICIAN, COMPUTER PROGRAMMER

AUTHOR: DENNIS OCHOSKI

DATE: AUGUST 2004   **PREVIOUS OUTLINE DATED:** AUGUST 2003

APPROVED:

_____   _____
DEAN   DATE

TOTAL CREDITS: 5

PREREQUISITE(S): CSD101

HOURS/WEEK: 4

Introduction to Object-oriented
Programming

I.   COURSE DESCRIPTION:

This course introduces students to the concepts of Object-Oriented
Programming and applies them in practical problem-solving exercises.
The course presently uses Microsoft Visual C++ 6.0 as the development
environment. Previous courses, CSD100 and CSD101, have developed
basic skills in C++ programming.

II.   LEARNING OUTCOMES AND ELEMENTS OF THE PERFORMANCE:

1.   **Demonstrate the use of structures and files as they relate to one
another. (chapters 10 and 11)**

Potential Elements of the Performance:

- Apply abstract data types (ADT) by creating the concept of a
structure.
- Develop and manipulate an array of structures.
- Use structures as parameters in function calls.
- Use hierarchical (nested) structures.
- Recall the basic concept of files.
- Understand and use binary files.
- Understand and use random access files.
- Write and debug programs utilizing the components above.

This learning outcome constitutes approximately 25% of the course.

2.   **Identify the components that form Object-oriented programming
(OOP) paradigm. (lecture notes)**

Potential Elements of the Performance:

- Identify the most important features of Object-oriented
programming languages.
- Assess the strengths and weaknesses of OOP and procedural
programming.
- Compare various OOP development environments.
- Compare C++ with Java.

This learning outcome constitutes approximately 2% of the course.

Introduction to Object-oriented
Programming

3. **Demonstrate the application of classes, encapsulation and polymorphism. (chapter 9: pgs. 450-459, chapters 11 and 13)**

   Potential Elements of the Performance:

   - Declare and use the C++ string class
   - Define classes and implement class members and member functions using appropriate encapsulation (data hiding) mechanisms.
   - Declare and define constructors and destructors for classes.
   - Use the *this* pointer to point to the invoking object.
   - Use dynamic memory allocation and de-allocation with *new* and *delete.*
   - Implement various Abstract Data Types (ADT), including a linked list, a stack and a queue.
   - Declare and define instance and static members.
   - Implement friend functions and friend classes appropriately.
   - Write and debug programs utilizing the components above.

   This learning outcome constitutes approximately 23% of the course.

4. **Write programs that use overloading, inheritance and exception handling to demonstrate object-oriented design. (chapter 14)**

   Potential Elements of the Performance:

   - Implement operator overloading using friend or member functions.
   - Use class inheritance to implement *Is-a* relationships and understand the difference between *composition and inheritance.*
   - Use virtual functions (late binding) to redefine class methods for derived classes.
   - Implement *Has-a* relationships with member objects (composition) or with private inheritance.
   - Use function and class templates to create related functions and classes.
   - Use C++ exception handling in programs.
   - Create a Win32 program using the MFC and Visual C++ (time permitting).

   This learning outcome constitutes approximately 25% of the course.

Introduction to Object-oriented   CSD205
Programming

III. TOPICS:

1. Additional components of structures and files.

2. Overview of the elements and objectives of OOP.

3. Classes, encapsulation, constructors, destructors, parameter passing.

4. Friend functions and operator overloading.

5. Inheritance, polymorphism and exception handling.

6. Templates

IV. REQUIRED RESOURCES/TEXTS/MATERIALS:

Textbook: "Brief Version of Starting Out With C++", 4th Edition, by Tony Gaddis, Scott Jones Publishers, 2004 ISBN 1-57676-121-5

V. EVALUATION PROCESS/GRADING SYSTEM:

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

Outcome	Quizzes	Assignments	Total
outcome #1:	15%	10%	25%
outcome #2, #3:	15%	10%	25%
outcome #3:	15%	10%	25%
outcome #4:	15%	10%	25%
	60%	40%	100%

(The percentages shown above may vary if circumstances warrant.)

NOTE: *It is required to pass both the theory and the assignment part of this course. It is not possible to pass the course if a student has a failing average in the four written quizzes but is passing the assignment portion, (or vice versa).*

QUIZZES

Quizzes will be announced about one week in advance. A zero grade will be given for quizzes missed without a valid reason given in advance.

Generally the only valid reasons are medical ones. Re-writes on these quizzes will not generally be possible so lab attendance is essential. All assigned work must be completed on time and quizzes must be written at the required time.

ASSIGNMENTS

An overall average grade of 60% must be achieved on the assignments pass the course. A late penalty will be applied when assignments are not submitted by the due date. A penalty of 10% per day may be applied to overdue assignments. After assignments have been handed back to the class (generally within one week), a grade of zero may be applied to overdue assignments. If lateness is due to extenuating circumstances, it is the student's responsibility to discuss the reasons for being late with the professor before the assignment is due.

The following semester grades will be assigned to students in this course:

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)	49% and below	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 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.	

Introduction to Object-oriented
Programming

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. If attendance is poor, eligibility for an X-grade is forfeited.

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 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.
While it is expected that students discuss assignments with each other and share ideas, it is not acceptable that students hand in work done by someone else and claim it as their own. Plagiarism on assignments will result in a zero grade being assigned for that assignment for everyone involved.

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.