

SAULT COLLEGE OF APPLIED ARTS & TECHNOLOGY  
SAULT STE. MARIE, ONTARIO

COURSE OUTLINE

COURSE TITLE: ADVANCED APPLICATION PROGRAMMING  
\_\_\_\_\_  
\_\_\_\_\_  
CODE NO.: EDP229-6 SEMESTER: FOUR  
\_\_\_\_\_  
\_\_\_\_\_  
PROGRAM: COMPUTER PROGRAMMER  
\_\_\_\_\_  
\_\_\_\_\_  
AUTHOR: DENNIS OCHOSKI  
\_\_\_\_\_  
\_\_\_\_\_  
DATE: JANUARY 1995  
\_\_\_\_\_  
\_\_\_\_\_  
PREVIOUS OUTLINE DATED: JANUARY, 1994  
\_\_\_\_\_  
\_\_\_\_\_

New:     <sup>x</sup>      Revision:     

APPROVED: \_\_\_\_\_  
DEAN, SCHOOL OF BUSINESS & HOSPITALITY  
DATE \_\_\_\_\_

Length of Course: 5 periods per week for one semester

Required Resources:

Text: C++ by Example, by Greg Perry

Disks: 2, 3 1/2" floppy diskettes

Philosophy/Goals:

This course will provide students with an opportunity to develop their programming skills using a "leading-edge" language, C++. C++ is expected to emerge as the dominant programming language of the mid-to-late 1990s. The course will re-emphasize the use of structured programming techniques and proper program design.

Special Notes:

1. Students are advised to maintain a copy of all files on a backup disk. Loss of an assignment due to a lost or damaged disk is not an acceptable reason for a late or incomplete assignment.
2. Students with special needs, due to such things as physical limitations, visual and/or hearing impairments, or learning disabilities, are encouraged to discuss required accommodations, confidentially, with the instructor.
3. There will be no re-writes in this course except in situations out of the control of the student (such as illness, urgent family matters, etc.) in which a re-write may be issued at the discretion of the instructor.
4. Assignments received after the due date are subject to grade of zero except in situations as specified in #3 above.

ADVANCED PROGRAMMING LANGUAGES

EDP229

COURSE NAME

COURSE CODE

Student Evaluation:

The student's final grade will consist of the following components:

Quizzes:

|                         |            |     |
|-------------------------|------------|-----|
| Quiz #1 - Modules 1 & 2 | 15%        |     |
| Quiz #2 - Module 3      | 15%        |     |
| Quiz #3 - Module 4      | 10%        |     |
| Quiz #4 - Module 5      | 15%        |     |
| Quiz #5 - Module 6      | <u>10%</u> |     |
|                         |            | 65% |

Assignments:

|                         |           |      |
|-------------------------|-----------|------|
| Asgn #1 - Modules 1 & 2 | 5%        |      |
| Asgn #2 - Module 3      | 5%        |      |
| Asgn #3 & #4 - Module 4 | 10%       |      |
| Asgn #5 & #6 - Module 5 | 10%       |      |
| Asgn #7 - Module 6      | <u>5%</u> |      |
|                         |           | 35%  |
|                         |           | 100% |

Grading:

|    |                           |
|----|---------------------------|
| A+ | 90 - 100%                 |
| A  | 80 - 89%                  |
| B  | 70 - 79%                  |
| C  | 60 - 69%                  |
| R  | UNDER 60% - Repeat Course |

Course Objectives:

1. Build upon programming skills acquired in previous semesters.
2. Learn a systems implementation language of choice in industry.
3. Further develop structured programming techniques and design.
4. Develop solutions to business information problems using C++.

ADVANCED PROGRAMMING LANGUAGES

EDP229

---

COURSE NAME

---

COURSE CODE

Module 1: Introduction to C++ Programming  
(chapters 3 - 8)

Objectives: When this module is completed, the student should be able to:

1. understand the C++ program development environment.
2. become familiar with fundamental data types.
3. use arithmetic operators.
4. understand the precedence of arithmetic operators.
5. use simple input/output statements.

Module 2: Control Structures  
(chapters 9 - 15)

Objectives: When this module is completed, the student should be able to:

1. use the *if*, *if/else*, and *switch* selection structures to choose among alternative actions.
2. use the *while*, *do/while*, and *for* looping structures to execute statements in a program repeatedly.
3. use the increment, decrement, assignment, and logical operators.
4. use the *break*, *exit*, and *continue* program control statements.

---

COURSE NAME

---

COURSE CODE

Module 3: Functions  
(chapters 16 - 22)

Objectives: When this module is completed, the student should be able to:

1. understand how to construct programs modularly from small pieces called functions.
2. become familiar with the common math functions available in the C standard library.
3. write and use custom-designed functions.
4. understand the mechanisms used to pass information between functions.
5. apply simulation techniques using random number generation.

Module 4: Arrays and Pointers  
(chapters 23 - 27)

Objectives: When this module is completed, the student should be able to:

1. understand the array data structure.
2. understand the use of arrays to store, sort, and search lists and tables of values.
3. pass arrays to functions.
4. understand basic sorting techniques.
5. declare and manipulate multiple-subscript arrays.
6. use pointers in various capacities.
7. understand the class relationship among pointers, arrays, and strings.
8. declare and use arrays of strings.

ADVANCED PROGRAMMING LANGUAGES

EDP229

---

COURSE NAME

---

COURSE CODE

Module 5: Structures and Files  
(chapters 28 - 31)

Objectives: When this module is completed, the student should be able to:

1. create and use structures.
2. use the functions of the string handling library (string.h).
3. create, read from, write to, and update files.
4. become familiar with sequential access files.
5. perform input from, and output to, character string arrays.

Module 6: Introduction to Object-Oriented Programming  
(chapter 32)

Objectives: When this module is completed, the student should be able to:

1. understand and create classes.
2. understand how to create, use, and destroy class objects.
3. control access to object data members and member functions.
4. appreciate the value of object orientation.