

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



COURSE OUTLINE

Course Title: COMPUTER PROGRAMMING I

Code No.: CSD100

Semester: FALL 1999

Program: PROGRAMMER(2090)/PROGRAMMER ANALYST(2091)

Instructor: DENNIS OCHOSKI

Date: SEPTEMBER 1999

Previously

Dated: SEPTEMBER 1998

Approved: _____

Dean

Date

COMPUTER PROGRAMMING I

CSD100

COURSE NAME

COURSE CODE

TOTAL CREDITS: 4

PREREQUISITE(S): None

I. COURSE DESCRIPTION:

This course is intended to provide a firm foundation of computer programming skills needed in the computer studies area. It is the first of two courses that use the C/C++ programming language to develop the student's computer programming and problem solving skills.

II. TOPICS TO BE COVERED:

1. Introduction to computer programming concepts.
2. Basic C/C++ program structure.
3. Input/output in C/C++.
4. Decisions/Conditions in C/C++.
5. Repetition/Looping in C/C++.
6. Modularization using User-Defined Functions

III. LEARNING OUTCOMES AND ELEMENTS OF THE PERFORMANCE:

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

1. Discuss and apply the concepts involved in the development of software to solve problems using the computer. (Unit 1 - Perry and lecture notes)

This learning outcome will comprise **15%** of the course.

Elements of the performance:

- define the concept of a "computer program/software"
- differentiate between prewritten software and custom-designed software
- differentiate between high level languages and machine language
- describe the top-down process of developing a program
- understand the "golden rule" for writing computer programs
- describe the purpose of a compiler/interpreter
- describe the process of transforming a source program to an executable module
- differentiate between batch processing and online processing
- write algorithms and describe them using pseudocode and flowcharts

2. Write a simple C/C++ program applying the concepts of program structure, arithmetic, and assignment. (Units 2, 3, 4, 5, 6: pgs. 90-105, 7 and 9: pgs. 183-187)

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

Elements of the performance:

- explain the main components of a C/C++ program
- name and distinguish C/C++'s basic data types
- explain and properly use the naming conventions for C/C++ identifiers
- understand and apply simple output statements using the *cout* operator
- differentiate between character, string, and numeric constants
- differentiate between character and numeric variables
- declare and initialize variables correctly
- explain computer memory concepts and how they relate to processing data

Elements of the performance(cont'd):

- use assignment operators (=, +=, -=, *=, /=, ++, --) for character and numeric data
- use the *strcpy()* function to assign string values to character variables
- use arithmetic operators and apply their precedence (+, -, *, /, %)
- evaluate integer and mixed-mode arithmetic correctly
- explain automatic promotion and apply typecasting to define data types
- differentiate between syntax and logic errors
- write and compile a simple program in C/C++ incorporating the concepts above

3. Develop algorithms and write C/C++ programs to solve problems involving the standard computer operations of input and output.
(Unit 6 pgs. 106-114)

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

Elements of the performance:

- apply the *cin* operator to perform input of data
- apply the *cout* operator to perform output of data
- apply the *getline()* function to accept string values that include a space(s)
- apply the *setw()*, *setprecision()*, and *setf()* manipulators to format output on the screen
- explain the purpose of "include" files for the *cin* and *cout* operators
- write, test, and debug programs using the *cin* and *cout* operators

4. Develop algorithms and write C/C++ programs to solve problems involving the standard computer operations of decisions/conditions and selection.
(Units 8 and 14)

This learning outcome will comprise **25%** of the course.

Elements of the performance:

- describe the use of the relational and logical operators, and use them to write both simple and complex logical expressions (==, !=, <, <=, >, >=, !, &&, ||)

Elements of the performance(cont'd):

- describe the operation of the following C/C++ decision-making structures and use them in C/C++ programs:
 - a. *if...else*
 - b. *nested ifs*
 - c. *if...else if...else*
 - d. the *switch* statement
 - write algorithms to solve problems containing decision-making structures, and describe them using pseudocode and flowcharts
 - write, test, and debug programs containing selection structures
5. Develop algorithms and write C/C++ programs to solve problems involving the standard computer operations of looping and repetition.
(Units 11, 12, and 13)

This learning outcome will comprise **25%** of the course.

Elements of the performance:

- discuss the concept of repetition/looping in computer programs
- describe the operation of the following C/C++ repetition structures and use them in C/C++ programs:
 - a. *while*
 - b. *do...while*
 - c. *for*
 - d. *nested loops*
 - e. *break* and *continue* statements
- write algorithms to solve problems containing repetition structures, and describe them using pseudocode and flowcharts
- describe and correct an "infinite loop" problem
- write, test, and debug programs containing repetition structures

COMPUTER PROGRAMMING I

CSD100

COURSE NAME

COURSE CODE

6. Discuss and create elementary user-written functions.
(Units 16 pgs. 331-339)

This learning outcome will comprise **15%** of the course.

Elements of the performance:

- distinguish between the *calling* and the *called* functions
- understand the concept of *scope*
- distinguish between *local* and *global* variables
- write, test, and debug programs containing functions

IV. EVALUATION METHODS:

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

Quizzes:

outcome #1	10%
outcomes #2 & #3	15%
outcome #4	20%
outcome #5	20%
outcome #6	<u>10%</u>
	75%

Assignments:

outcome #1	5%
outcomes #2 & #3	5%
outcome #4	5%
outcome #5	5%
outcome #6	<u>5%</u>
	25%

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	< 60%	Repeat the course
X	Incomplete. A temporary grade limited to special circumstances 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.	

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. There will be **no** supplemental or make-up quizzes/tests at the end of the semester.
3. 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.
4. Any assignment submissions deemed to be copied will result in a **zero** grade being assigned to **all** students involved in that particular incident.
5. 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.
6. Students with special needs (eg. physical limitations, visual impairments, hearing impairments, learning disabilities) are encouraged to discuss required accommodations confidentially with the instructor.
7. 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 STUDENT RESOURCES

Text: Programming C++ in 12 Easy Lessons
 by Greg Perry

Diskettes: minimum of 3, 3 1/2"