

SAULT COLLEGE OF APPLIED ARTS AND TECHNOLOGY

SAULT STE. MARIE, ONTARIO



Sault College

COURSE OUTLINE

COURSE TITLE: COMPUTER PROGRAMMING I

CODE NO. : CSD100 **SEMESTER:** 1

PROGRAM: ALL COMPUTER STUDIES PROGRAMS

AUTHOR: Tycho Black, Dennis Ochoski, Bazlur Rasheed

DATE: August, 2000 **PREVIOUS OUTLINE DATED:** Sept, 1999

APPROVED:

DEAN DATE
TOTAL CREDITS: 5

PREREQUISITE(S): NONE

HOURS/WEEK: 5

Copyright ©2000 The Sault College of Applied Arts & Technology
Reproduction of this document by any means, in whole or in part, without prior written permission of Sault College of Applied Arts & Technology is prohibited.
For additional information, please contact Kitty DeRosario,
School of Trades & Technology
(705) 759-2554, Ext. 642

I. COURSE DESCRIPTION:

This course provides a firm foundation of computer programming skills needed in the computer studies programs. 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. 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.**

ELEMENTS OF THE PERFORMANCE

- define the concept of a "computer program/software"
- differentiate between high level languages and machine language
- describe the history of programming languages and C/C++ in particular
- describe the top-down process of developing a program
- describe the purpose of a compiler/interpreter
- explain computer memory concepts and how they relate to processing data
- describe the process of creating a source program and transforming it into an executable module
- differentiate between batch processing and interactive systems
- write algorithms and describe them using pseudocode and flowcharts
- compare and utilize various development environments for C++ programs

This learning outcome will comprise 15% of the course. (Ref: Chap.1 Dale)

- 2. Write simple C/C++ programs applying the concepts of program structure, arithmetic, and assignment.**

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 stream output with *cout*
- differentiate between character, string, and numeric constants
- differentiate between character and numeric variables
- declare and initialize variables correctly
- use assignment operators (=, +=, -=, *=, /=, ++, --) for character and numeric data
- use arithmetic operators and apply their precedence (+, -, *, /, %)
- evaluate integer and mixed-mode arithmetic correctly
- use other C++ operators appropriately, including bitwise, conditional and sizeof() operators
- 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

This learning outcome will comprise 15% of the course. (Ref: Chap.2, 3 and 10 Dale)

3. Develop algorithms and write C/C++ programs to solve problems involving the standard computer operations of input and output.

ELEMENTS OF THE PERFORMANCE

- apply stream input using *cin* to perform input of data
- apply stream output using *cout* and various manipulators to perform formatted output of data
- apply the *getline()* function to accept string values that include a space(s)
- write, test, and debug programs using stream input/output
- write, test, and debug programs using C-type standard input/output with *printf()* and *scanf()*.

This learning outcome will comprise 20% of the course. (Ref: Chap 4 Dale)

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

ELEMENTS OF THE PERFORMANCE

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

This learning outcome will comprise 20% of the course. (Ref: Chap 5 Dale)

5. Develop algorithms and write C/C++ programs to solve problems involving the standard computer operations of looping and repetition.

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

This learning outcome will comprise 20% of the course. (Ref: Chap 6 and 9 Dale)

6. Discuss and create elementary user-written functions.

ELEMENTS OF THE PERFORMANCE

- understand the role and operation of functions in C/C++
- 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

This learning outcome will comprise 10% of the course. (Ref: Chap 7 Dale)

III. TOPICS:

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

IV. REQUIRED RESOURCES/TEXTS/MATERIALS:

Textbook:

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

V. EVALUATION PROCESS/GRADING SYSTEM:

3 Written Tests @ 20% each	60%
Lab assignments/Lab attendance/Lab Quizzes	<u>40%</u>
	100%

ELIGIBILITY FOR XGRADES/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:

1. The student's attendance has been satisfactory.
2. An overall average of at least 50% has been achieved.
3. The student has not had a failing grade in all of the theory tests taken.
4. 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.

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:

<u>Grade</u>	<u>Definition</u>	<u>Grade Point Equivalent</u>
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 <i>Policies & Procedures Manual – Deferred Grades and Make-up</i>).	
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.	

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 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.

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.