Douglas College

EFFECTIVE: SEPTEMBER 2003
CURRICULUM GUIDELINES

A. Division: **Instructional**

Effective Date: SEPTEMBER 2003

B. Department / Program Area: **Computing Science**

Revision: ☒ New Course: □

If Revision, Section(s) Revised: **H, J, M, N, O, P, Q**

Date of Previous Revision: **February 7, 2002**

Date of Current Revision: **November 18, 2002**

C: CMPT 110	**D: Introduction to Computing Science Using C++**	**E: 4**
Subject & Course No.	Descriptive Title	Semester Credits

F: Calendar Description:

This course introduces the science of computing. Emphasis is placed on the analysis of problems, the design of algorithms, and the abstraction of control and data in computer implementations of the design. Initially structured top-down design and procedural programming is used followed by an introduction to recursive functional programming and an introduction to Object Oriented Design (OOD) and Object Oriented Programming (OOP). C++ is used as the implementation language.

G: Allocation of Contact Hours to Type of Instruction / Learning Settings

Primary Methods of Instructional Delivery and/or Learning Settings:

Lecture / Laboratory

Number of Contact Hours: (per week / semester for each descriptor)

Lecture **4 hours / week**
Laboratory **2 hours / biweekly**

Number of Weeks per Semester: **15**

H: Course Prerequisites:
CMPT 101 with a minimum grade of C and MATH 110 with a minimum grade of C; or CMPT 101 with a minimum grade of C and BC Principles of Math 12 with a minimum grade of B

I: Course Corequisites:
None

J: Course for which this Course is a Prerequisite:
CMPT 150 and CMPT 210

K: Maximum Class Size:
Lecture **34**
Laboratory **34**

L: PLEASE INDICATE:

□ Non-Credit
□ College Credit Non-Transfer
☒ College Credit Transfer:

SEE BC TRANSFER GUIDE FOR TRANSFER DETAILS (www.bccat.bc.ca)

M: Course Objectives / Learning Outcomes:

Students should be able to

- **Analyze problem specification**
- **Design, using either a structured top-down methodology or OOD to solve a well defined problem**
- **Implement a program from a design using either a structured programming or OOP**
- **Document a project**

Students should understand the concepts of

- **Programming cycle**
- **Software life cycle and the importance of reusability and maintainability**
- **Generality through abstractions and maintainability, reusability, and extensibility through modularity**

Students should have experience with

- **Standard control structures including sequence, selection, and iteration**
- **Functions (void and value returning), parameter passing (by value and by reference), and recursion**
- **One-dimensional array**

N: Course Content:

1. **Program structure**
2. **Primitive data types, operators and expressions**
3. **Control structures**
   - **3.1. Selections**
   - **3.2. Repetitions**
4. **Strings**
5. **Program input and output**
   - **5.1. Standard input/output devices**
   - **5.2. External files**
6. **Built-in functions**
7. **Functions and parameter passing**
   - **7.1. Value and reference parameters**
   - **7.2. Scope and lifetime of identifiers**
8. **Recursive functions**
9. **One-dimensional arrays**
10. **Introduction to pointers**
    (domain of arrays and parameter passing)
11. **Structures (records)**
12. **OOD and OOP**
    - **12.1. Separate compilation**
    - **12.2. Introduction to inheritance**

O: Methods of Instruction:

There are three components to the course: lectures, labs, and self directed learning (i.e. programming assignments)

The lecture is used to introduce new material, usually via a sequence of theoretical concepts and examples. The textbook is to be used as an additional source of study material, problems, and examples.

The two-hour biweekly lab is exclusively used to evaluate the student's practical programming ability.

Assignments are marked according to correctness of the algorithms, efficiency, and programming style.

P: Textbooks and Materials to be Purchased by Students:

- **Dale N., Weems C., Headington M., Programming and Problem Solving with C++, Third Edition, D.C. Heath and Company**
- **Portfolio for Programming Assignments**
- **Two 3 ½ " high density diskettes**

Q: Means of Assessment:

Evaluation will be carried out in accordance with Douglas College policy. The instructor will present a written course outline with specific evaluation criteria at the beginning of semester. Evaluation will be based on some of the following:

labs (6 to 7)	**15% - 25%**
assignments (4 to 6)	**20% - 30%**
tests (1 to 2) @ 15% - 30% each	**15% - 60%**
final examination	**25% - 40%**
class participation[1]	**0% - 5%**

Note #1: participation includes (but is not limited to) short pop-quizzes and/or attendance

R: Prior Learning Assessment and Recognition: specify whether course is open for PLAR

Not at this time

Course Designer(s):

Education Council / Curriculum Committee Representative:

Dean / Director:

Registrar;