Douglas College

EFFECTIVE: MAY 2003
CURRICULUM GUIDELINES

A. Division: **Instructional**

Effective Date: May 2003

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

Revision: ☒ New Course: □

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

Date of Previous Revision:

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

C: CMPT 210	**D: Data and Control Structures**	**E: 4**
Subject & Course No.	Descriptive Title	Semester Credits

F: Calendar Description:

This course continues the study of Object Oriented Design (OOD) and Object Oriented Programming (OOP) with a study of inheritance and polymorphism. Other topics include an introduction to the analysis of algorithms, techniques for searching state spaces, and dynamic data structures including lists, stacks, queues, and trees. Programs are written in C++.

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: **14**

H: Course Prerequisites:
CMPT 110 with a minimum grade of C

Note: MATH 130 is highly recommended as a prerequisite

I: Course Corequisites:
None

J: Course for which this Course is a Prerequisite:
None

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 understand the concepts of

- **Inheritance**
- **Dynamic versus static data structures**
- **Late/dynamic binding and polymorphism**
- **Asymptotic behavior of algorithms**

Student should be able to

- **Analyze the time complexity of iterative and recursive algorithms**
- **Use OOD on problems where inheritance is advantageous**
- **Take advantage of polymorphism**
- **Choose the most appropriate abstract data structure and be able to implement it efficiently**

N: Course Content:

1. Modules, information hiding and inheritance

2. Analysis of algorithms (best case, worst case, average case)

2.1. Search algorithms – hashing, sequential and binary search

2.2. Sort algorithms – bubble, selection, linear insertion, binary insertion, mergesort, quicksort

3. Dynamic data structures

3.1. Linear structures – lists, stacks, queues

3.2. Trees

3.2.1.1. Binary trees
Recursive algorithms for tree traversals
Iterative algorithms for searching a tree (depth-first using a stack, breadth-first using a queue, and heuristic using a priority queue)
Binary search trees
Expression trees
Tree sort

3.2.1.2. Heaps
Heap sort
Priority queue

Optional

- **Trie**
- **Huffman codes**

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:

- **Headington M., Riley D., Data Abstraction and Structures Using C++, 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: