Creation date: March 24, 2021
Revision date:

**SAMPLE COURSE OUTLINE**

**Course Code, Number, and Title:**

CPSC 1160: Algorithms and Data Structures I

**Course Format:**
[Course format may vary by instructor. The typical course format would be:]

Lecture 4.0 h + Seminar 0.0 h + Lab. 2.0 h

**Credits:** 3.0 **Transfer Credit:** For information, visit bctransferguide.ca

**Course Description, Prerequisites, Corequisites:**

Students focus on practicing and developing programing skills. Students implement algorithms along with appropriate data structures to produce good software. Students apply recursion, abstract data types, algorithm analysis, sorting and searching algorithms, pointers, arrays, dynamic memory management, linked lists, stacks, and queues. Students also learn about low-level data representations and systematic software development. As a tool, object-oriented programming is introduced.

Prerequisite(s): A minimum "C" grade in one of CPSC 1150 or 1155; and one of the following: a minimum "B" grade in Precalculus 12; or a minimum "C" grade in MATH 1170, 1171, 1173, or 1174; or a minimum "C+" in Precalculus 12 and a minimum "C-" grade in Calculus 12; or MDT 85. Prerequisites are valid for only three years.

**Learning Outcomes:**

Upon successful completion of this course, students will be able to…

- Estimate the running time of a given algorithm
- Design and implement efficient (in terms of time complexity) recursive and non-recursive algorithms
- Implement an abstract data type as a class
- Analyze and implement searching and comparative sorting algorithms on linear data structures (sequential search, binary search, selection sort, insertion sort, bubble sort, quick sort, and merge sort)
- Design and implement linear data structures (vectors, matrices, singly linked lists, doubly linked lists, circular linked lists, stacks and queues) and an introduction of their associated iterators where applicable
- Find and use appropriate libraries, resources and documentation
- Use good software development principles (modularization, maintainability, documentation, memory management and testing)
- Implement generic functions and classes

**Instructor(s): TBA**
**Office: TBA** **Phone: (604) 323-XXXX**
**Office Hours: TBA** **Email: TBA**

**Textbook and Course Materials:**
[Textbook selection may vary by instructor. An example of texts and course materials for this course might be:]

For textbook information, visit https://mycampusstore.langara.bc.ca/buy_courselisting.asp?selTerm=3|8

Note: *This course may use an electronic (online) instructional resource that is located outside of Canada for mandatory graded class work. You may be required to enter personal information, such as your name and email address, to log in to this resource. This means that your personal information could be stored on servers located outside of Canada and may be accessed by U.S. authorities, subject to federal laws. Where possible, you may log in with an email pseudonym as long as you provide the pseudonym to me so I can identify you when reviewing your class work.*

**Assessments and Weighting:**
**Final Exam** 20%
**Other Assessments** 80%
[An example of other assessments might be:]

Best 9 out of 10 Assignments 45%
Two Code Reviews of the Assignments 10%
Labs 5%
Two Midterms 20%

**Grading System:**
Specific grading schemes will be detailed in each course section outline.

*Information unavailable, please consult Department for details.*

**Topics Covered:**
[Topics covered may vary by instructor. An example of topics covered might be:]

- C++ basics, data types, C++ strings & I/O in C++
- C++ arrays & functions, parameter passing & random numbers
- Big manipulation and recursion
- Sorting: selection, insertion, bubble sort quicksort, mergesort
- Complexity analysis, structs & enumeration type
- Abstract Data Types & C++ classes
- Object oriented programming in C++
- Operator overloading, pointers and dynamic memory
- Dynamic arrays, vector class & shallow vs. deep coding
- Copy constr., overloaded =, destructor
- Linked lists & stack implementations
- Queues & linked list sets

- Statis variables/functions/constants
- C++ templates & the STL
- Class templates & binary search trees
- Hashing & error handling (exceptions)

As a student at Langara, you are responsible for familiarizing yourself and complying with the following policies:

**College Policies:**

E1003 - Student Code of Conduct
F1004 - Code of Academic Conduct
E2008 - Academic Standing - Academic Probation and Academic Suspension
E2006 - Appeal of Final Grade
F1002 - Concerns about Instruction
E2011 - Withdrawal from Courses

**Departmental/Course Policies:**

*Information unavailable, please consult Department for details.*