ORIGINAL COURSE IMPLEMENTATION DATE:	November 1999
REVISED COURSE IMPLEMENTATION DATE:	September 2016
COURSE TO BE REVIEWED: (six years after UEC approval)	January 2022

Course outline form version: 09/15/14

OFFICIAL UNDERGRADUATE COURSE OUTLINE FORM

Note: The University reserves the right to amend course outlines as needed without notice.

Course Code and Number: COMP 251	**Number of Credits:** 4 Course credit policy (105)
Course Full Title: Data Structures and Algorithms **Course Short Title (if title exceeds 30 characters):**	
Faculty: Faculty of Professional Studies	**Department (or program if no department):** Computer Information Systems

Calendar Description:

An introduction to the abstract data structures used in the solution of common computing problems. Applies concepts such as stacks, queues, trees, and graphs to problems using a modern object-oriented programming language. Also covers the correctness and efficiency of algorithms.

Prerequisites (or NONE):	COMP 155 and MATH 125. Note: As of January 2017, prerequisites will change to the following: COMP 125, COMP 155, and MATH 125.
Corequisites (if applicable, or NONE):	
Pre/corequisites (if applicable, or NONE):	

Equivalent Courses (cannot be taken for additional credit)

Former course code/number: **COMP 175**

Cross-listed with:

Equivalent course(s):

Note: Equivalent course(s) should be included in the calendar description by way of a note that students with credit for the equivalent course(s) cannot take this course for further credit.

Transfer Credit

Transfer credit already exists: ☒ Yes ☐ No

Transfer credit requested (OReg to submit to BCCAT):
☐ Yes ☒ No (if yes, fill in transfer credit form)

Resubmit revised outline for articulation: ☐ Yes ☒ No

To find out how this course transfers, see **bctransferguide.ca**.

Total Hours: 60

Typical structure of instructional hours:

Lecture hours	45
Seminars/tutorials/workshops	
Laboratory hours	15
Field experience hours	
Experiential (practicum, internship, etc.)	
Online learning activities	
Other contact hours:	
Total	**60**

Special Topics

Will the course be offered with different topics?
☐ Yes ☒ No

If yes, different lettered courses may be taken for credit:
☐ No ☐ Yes, repeat(s) ☐ Yes, no limit

Note: The specific topic will be recorded when offered.

Maximum enrolment (for information only): 35

Expected frequency of course offerings (every semester, annually, every other year, etc.): 1 to 2 times per year

Department / Program Head or Director: Dan Harris	**Date approved:**	Sept 25, 2015
Faculty Council approval	**Date approved:**	October 16, 2015
Campus-Wide Consultation (CWC)	**Date of posting:**	November 20, 2015
Dean/Associate VP: Lucy Lee	**Date approved:**	October 16, 2015
Undergraduate Education Committee (UEC) approval	**Date of meeting:**	January 29, 2016

Learning Outcomes

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

- Describe the following abstract data types:
  - Stacks
  - Lists
  - Queues
  - Arrays
  - Heaps
  - Trees
  - Dictionaries
- Implement these abstract data types in an object-oriented programming language.
- Calculate space and time complexity for commonly used algorithms.
- Identify which algorithms are appropriate for a given problem.
- Identify which data structures are appropriate for a given problem.

Prior Learning Assessment and Recognition (PLAR)

☒ Yes ☐ No, PLAR cannot be awarded for this course because

Typical Instructional Methods (guest lecturers, presentations, online instruction, field trips, etc.; may vary at department's discretion)

The course will be delivered in lecture-lab format, with numerous demonstrations and hands-on activities. The lab portion gives students and the instructor the ability to view and interact with current projects.

Grading system: Letter Grades: ☒ Credit/No Credit: ☐ Labs to be scheduled independent of lecture hours: Yes ☐ No ☒

NOTE: The following sections may vary by instructor. Please see course syllabus available from the instructor.

Typical Text(s) and Resource Materials (if more space is required, download Supplemental Texts and Resource Materials form)

	Author (surname, initials)	Title (article, book, journal, etc.)	Current ed.	Publisher	Year
1.	Dale, Joyce, Weems	Object-Oriented Data Structures Using Java	☒	Jones and Bartlett	2011
2.			☐		
3.			☐		
4.			☐		
5.			☐		

Required Additional Supplies and Materials (software, hardware, tools, specialized clothing, etc.)

None

Typical Evaluation Methods and Weighting

Final exam:	45%	Assignments:	30%	Midterm exam:	25%	Practicum:	%
Quizzes/tests:	%	Lab work:	%	Field experience:	%	Shop work:	%
Other:	%	Other:	%	Other:	%	Total:	100%

Details (if necessary):

Typical Course Content and Topics

- Software Engineering
- Data Design & Implementation
- Abstract Data Types
- Unsorted Lists & Sorted Lists
- Stacks and Queues
- Linked Structures
- Recursion
- Binary Search Trees
- Priority Queues, Heaps, & Graphs
- Hashing
- Maps
- Efficiency of Algorithms
- Sorting & Searching