UNIVERSITY OF THE FRASER VALLEY

ORIGINAL COURSE IMPLEMENTATION DATE: November 1999
REVISED COURSE IMPLEMENTATION DATE: September 2026
COURSE TO BE REVIEWED (six years after UEC approval)**:** March 2032
Course outline form version: 29/08/2024

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:** Data Structures & Algorithms	
Faculty: Faculty of Business and Computing	**Department/School:** Computing

Calendar Description:

Focuses on essential concepts in data structures and algorithms including arrays, linked-lists, stacks, queues, priority queues, dynamic arrays, heaps, search trees, hash tables and graphs. The curriculum introduces classical algorithmic strategies for sorting, searching, and recursion using modern object-oriented programming.

Prerequisites (or NONE):	COMP 125, COMP 155, and MATH 125.
Corequisites (if applicable, or NONE):	
Pre/corequisites (if applicable, or NONE):	

Antirequisite Courses *(Cannot be taken for additional credit.)*

Former course code/number:

Cross-listed with:

Equivalent course(s):

(If offered in the previous five years, antirequisite course(s) will be included in the calendar description as a note that students with credit for the antirequisite course(s) cannot take this course for further credit.)

Typical Structure of Instructional Hours

Lecture/seminar	45
Supervised laboratory hours (computer lab)	15
Total hours	**60**

Scheduled Laboratory Hours

Labs to be scheduled independent of lecture hours: **No**

Course Details

Special Topics course: **No**
(If yes, the course will be offered under different letter designations representing different topics.)

Directed Study course: **No**
(See policy 207 for more information.)

Grading System: **Letter grades**

Delivery Mode: **May be offered in multiple delivery modes**

Expected frequency: **Every semester**

Maximum enrolment (for information only)**: 35**

Prior Learning Assessment and Recognition (PLAR)

PLAR is available for this course.

Transfer Credit *(See **bctransferguide.ca**.)*

Transfer credit already exists: **Yes**

Submit outline for (re)articulation: **No**
(If yes, fill in transfer credit form.)

Department approval	**Date of meeting:**	May 2025
Faculty Council approval	**Date of meeting:**	September 12, 2025
Undergraduate Education Committee (UEC) approval	**Date of meeting:**	March 27, 2026

Learning Outcomes *(These should contribute to students' ability to meet program outcomes and thus Institutional Learning Outcomes.)*

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

1. Demonstrate a foundational understanding of essential concepts related to data structures and algorithms.
2. Design algorithmic strategies for solving classical computational problems, such as sorting, graph searching, and recursion.
3. Implement various algorithms and their corresponding data structures in a programming environment.
4. Analyze functionality and performance metrics for various algorithms and data structures.
5. Reflect on how Indigenous knowledge systems, such as Stó:lō seasonal cycles and resource management practices, illustrate principles of algorithmic thinking, including recursion, iteration, and optimization.

Recommended Evaluation Methods and Weighting *(Evaluation should align to learning outcomes.)*

Final exam:	30%	Quizzes/tests/midterm:	25%	Holistic assessment:	5%
Lab work:	10%	Assignments:	30%		%

Details:

Quizzes (5%) and midterm exam (20%)

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

Typical Instructional Methods *(Guest lecturers, presentations, online instruction, field trips, etc.)*

Structured lectures which include instructor presentations, in-class polling and collaborative learning such as group assignments and labs.

Texts and Resource Materials *(Include online resources and Indigenous knowledge sources. Open Educational Resources (OER) should be included whenever possible. If more space is required, use the Supplemental Texts and Resource Materials form.)*

	Type	Author or description	Title and publication/access details	Year
1.	Textbook	Michael T. Goodrich, Roberto Tamassia and Michael H. Goldwasser	Data Structures and Algorithms in Java, 6th Edition	2014
2.	Book	Mark Allen Weiss	Data Structures and Algorithm Analysis in C++, Pearson	2011
3.	Book	Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein	Introduction to Algorithms (No specific programming language -- it uses "Pseudocode"), MIT press	2009
4.	Indigenous knowledge	Keith Thor Carlson	You are asked to Witness: The Stó:lō in Canada's Pacific Coast History.	1997
5.	OER	Pat Morin	Open Data Structures in Java, access at BCcampus	2011

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

Recommended Integrated Development Environment (IDE) is: (i) BlueJ or (ii) Eclipse.

Course Content and Topics

- Fundamental data structures algorithms analysis
- Stacks, queues, and deques
- Trees
- Priority queues and heaps
- Recursion
- Maps and hash tables
- Search trees
- Sorting algorithms
- Graphs
- Indigenous resource management practices and algorithmic thinking