UNIVERSITY OF THE FRASER VALLEY

ORIGINAL COURSE IMPLEMENTATION DATE: September 2015
REVISED COURSE IMPLEMENTATION DATE: September 2026
COURSE TO BE REVIEWED (six years after UEC approval)**:** March 2032
Course outline form version: 28/10/2022

OFFICIAL UNDERGRADUATE COURSE OUTLINE FORM

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

Course Code and Number: COMP 359	**Number of Credits:** 3 Course credit policy (105)
Course Full Title: Design and Analysis of Algorithms **Course Short Title:** Design & Analysis of Algorithms	
Faculty: Faculty of Business and Computing	**Department/School:** School of Computing

Calendar Description:

Advanced study of the analysis and design of algorithms and data structures techniques with a focus on problem-solving strategies, algorithm efficiency, correctness, and performance metrics through group coursework and communication through class presentations.

Prerequisites (or NONE):	COMP 251, (MATH 125 or MATH 225), and (STAT 106 or MATH 270/STAT 270).
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	30
Supervised laboratory hours (computer lab)	15
Total hours	45

Scheduled Laboratory Hours

Labs to be scheduled independent of lecture hours: ☒ No ☐ Yes

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: **Annually**

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: **No**

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

Department approval	**Date of meeting:**	April 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. Design algorithms for various problem types using different algorithmic and data structure paradigms.
2. Analyze algorithm performance using asymptotic notations to measure their time and space complexities.
3. Implement efficient algorithms for solving real-world computational problems with reference to Indigenous-applied case-studies.
4. Collaborate to document and clearly communicate implementations using statistical experiments that demonstrate precision.
5. Reflect on their personal experience of collaboration in problem solving involving Indigenous group-decision-making practices.

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

Project:	60%	Assignments:	30%	Holistic assessment:	10%
	%		%		%

Details:

- Project 1 (presentation of assignment 1), 15%
- Project 2 (presentation of assignment 2), 20%
- Project 3 (presentation of assignment 3), 25%

Each project is completed as a presentation of and after its corresponding assignment. Each assignment and presentation are typically done in groups of three. The first two projects are given as screen-recorded videos shared on a discussion board. The last project is completed as an in-person presentation on-campus during the final exam period.

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.)*

Lectures including hands-on programming, and assignments

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	Anany Levitin	Introduction to the Design and Analysis of Algorithms	2011
2.	Textbook	Cormen, Leiserson, Rivest, and Stein	Introduction to Algorithms	2022
3.	OER book	Johan Sannemo	Principles of Algorithmic Problem Solving	2018
4.	Journal	Pinotti, T., Adler, M.A., Mermejo, R. *et al.*	Picuris Pueblo oral history and genomics reveal continuity in US Southwest. *Nature* (2025). https://doi.org/10.1038/s41586-025-08791-9	2025
5.				

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

Compilers or interpreters for a programming language such as Python, Java, C++, or C# and the associated pre-built data structures and algorithms Libraries. Software to visualize algorithm execution (e.g., Visualgo).

Course Content and Topics

- Review of data structure and simple algorithms
- Sorting and searching
- Directed and undirected graphs
- Decrease-and-conquer
- Divide-and-conquer
- Transform-and-conquer
- Algorithm analysis
- Randomized algorithms
- Dynamic programming
- Greedy algorithms
- Ethics in algorithm design and intersections with indigenous perspectives
- Indigenous group decision-making (consensus; circle method; roles of elder, matriarch, youth)