ORIGINAL COURSE IMPLEMENTATION DATE:	May 1994
REVISED COURSE IMPLEMENTATION DATE:	September 2021
COURSE TO BE REVIEWED (six years after UEC approval):	February 2027

Course outline form version: 05/18/2018

OFFICIAL UNDERGRADUATE COURSE OUTLINE FORM

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

Course Code and Number: MATH 343

Number of Credits: 3 Course credit policy (105)

Course Full Title: Applied Discrete Mathematics

Course Short Title:

(Transcripts only display 30 characters. Departments may recommend a short title if one is needed. If left blank, one will be assigned.)

Faculty: Faculty of Science

Department (or program if no department): Mathematics & Statistics

Calendar Description:

Algorithms are studied with an emphasis on discrete math, rather than programming. In particular, this course will cover some standard algorithms in combinatorics, running time analysis, correctness of algorithms, and techniques for selecting an appropriate algorithm to solve a problem.

Prerequisites (or NONE):	One of MATH 225, MATH 221, or COMP 251.
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:

Dual-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 hours	50
Tutorials/workshops	
Supervised laboratory hours	
Experiential (field experience, practicum, internship, etc.)	
Supervised online activities	
Other contact hours:	
Total hours	**50**

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

Special Topics *(Double-click on boxes to select.)*

This course is offered with different topics:

☒ No ☐ Yes *(If yes, topic will be recorded when offered.)*

Independent Study

If offered as an Independent Study course, this course may be repeated for further credit: *(If yes, topic will be recorded.)*

☒ No ☐ Yes, repeat(s) ☐ Yes, no limit

Transfer Credit

Transfer credit already exists: *(See **bctransferguide.ca**.)*

☒ No ☐ Yes

Submit outline for (re)articulation:

☒ No ☐ Yes *(If yes, fill in transfer credit form.)*

Grading System

☒ Letter Grades ☐ Credit/No Credit

Maximum enrolment (for information only): 36

Expected Frequency of Course Offerings:

Every second year *(Every semester, Fall only, annually, etc.)*

Department / Program Head or Director: Ian Affleck	**Date approved:**	June 15 2020
Faculty Council approval	**Date approved:**	September 11, 2020
Dean/Associate VP:	**Date approved:**	September 11, 2020
Campus-Wide Consultation (CWC)	**Date of posting:**	February 5, 2021
Undergraduate Education Committee (UEC) approval	**Date of meeting:**	February 26, 2021

Learning Outcomes:

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

1. Implement algorithms by hand on small examples.
2. Use algorithms to solve standard combinatorial problems (searching, sorting, string matching, bin packing, vertex colouring) via various appropriate approaches.
3. Decide when to use a heuristic approach to produce approximate answers.
4. Identify and create combinatorial objects such as permutations and partitions.
5. Identify graph theoretical structures such as paths, cycles and trees.
6. Use appropriate data structures (such as arrays and binary trees) when implementing algorithms.
7. Analyze the average case and worst case complexity of an algorithm.
8. Model a problem and use an appropriate algorithm to solve the problem.
9. Prove the correctness of an algorithm.

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 primarily lecture-based..

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.)*

The textbook is chosen by a departmental curriculum committee. Recommended texts are:

	Author (surname, initials)	Title (article, book, journal, etc.)	Current ed.	Publisher	Year
1.	J. Kleinberg, E. Tardos	*Algorithm Design*	☐		2005
2.	A. Levitin	*The Design and Analysis of Algorithms*	☐		2011
3.			☐		
4.			☐		
5.			☐		

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

Typical Evaluation Methods and Weighting

Final exam:	40%	Assignments:	15%	Field experience:	%	Portfolio:	%
Midterm exam:	%	Project:	%	Practicum:	%	Other:	%
Quizzes/tests:	45%	Lab work:	%	Shop work:	%	Total:	100%

Details (if necessary):

A student must obtain at least 40% on the final exam in order to pass this course.

Typical Course Content and Topics

1. Concepts of combinatorics and graph theory: combinations, permutations, partitions, trees, paths and cycles
2. Computer representation of combinatorial objects
3. Sorting, searching, string matching and min/max algorithms
4. Running time analysis of algorithms: worst-case and average-case analysis, asymptotic orders of growth
5. Running time complexity classes: Polynomial (P), Non-Deterministic Polynomial (NP) , NP-complete (NP-c) and NP-hard
6. Heuristics and approximation algorithms
7. Bin packing, vertex cover and graph colouring algorithms
8. Greedy algorithms
9. Randomized algorithms