UNIVERSITY COLLEGE of the FRASER VALLEY

COURSE IMPLEMENTATION DATE:	May, 1994
COURSE REVISED IMPLEMENTATION DATE:	Sept, 2004
COURSE TO BE REVIEWED:	Sept, 2008
(Four years after implementation date)	(MMMM YY format)

OFFICIAL COURSE OUTLINE INFORMATION

Students are advised to keep course outlines in personal files for future use.

Shaded headings are subject to change at the discretion of the department and the material will vary - see course syllabus available from instructor

FACULTY/DEPARTMENT: **Science, Health and Human Services/ Mathematics and Statistics**

COURSE NAME/NUMBER	FORMER COURSE NUMBER	UCFV CREDITS
MATH 343		**3**

COURSE DESCRIPTIVE TITLE: **Applied Discrete Mathematics**

CALENDAR DESCRIPTION:

This course introduces discrete modeling. Topics covered include generation of combinatorial objects, applications to scheduling, and applications of graphs.

PREREQUISITES: **MATH 225 AND KNOWLEDGE OF A COMPUTING LANGUAGE ACCEPTABLE TO THE INSTRUCTOR.**

COREQUISITES:

SYNONYMOUS COURSE(S)

(a) Replaces: ____________ *(Course #)*

(b) Cannot take: ____________ *(Course #)* for further credit.

SERVICE COURSE TO:

____________ *(Department/Program)*

____________ *(Department/Program)*

TOTAL HOURS PER TERM: **60**

STRUCTURE OF HOURS:

Lectures:	**60**	Hrs
Seminar:		Hrs
Laboratory:		Hrs
Field Experience:		Hrs
Student Directed Learning:		Hrs
Other (Specify):		Hrs

TRAINING DAY-BASED INSTRUCTION

LENGTH OF COURSE: ____________

HOURS PER DAY: ____________

MAXIMUM ENROLLMENT:	**36**	
EXPECTED FREQUENCY OF COURSE OFFERINGS:	**every second year**	
WILL TRANSFER CREDIT BE REQUESTED? (lower-level courses only)	☐ Yes	☐ No
WILL TRANSFER CREDIT BE REQUESTED? (upper-level requested by department)	☐ Yes	☒ No
TRANSFER CREDIT EXISTS IN BCCAT TRANSFER GUIDE:	☐ Yes	☒ No

AUTHORIZATION SIGNATURES:

Course Designer(s):	Math Department	Chairperson:	Peter Mulhern *(Curriculum Committee)*
Department Head:	Gillian Mimmack	Dean:	Jackie Snodgrass
PAC Approval in Principle Date:		PAC Final Approval Date:	November 26, 2003

LEARNING OBJECTIVES / GOALS / OUTCOMES / LEARNING OUTCOMES:

This course is a survey of some combinatorial aspects of computing. Students study algorithms for enumeration, optimization, applications of graph theory and other discrete problems. Issues of complexity are discussed. Students are also exposed to some intermediate-level concepts of combinatorics..

This course is designed to enable students to:

1. become acquainted with some well known algorithms in the area of combinatorics and graph theory.
2. be able to implement the algorithms discussed in classes on computers.
3. become acquainted with computer representation of combinatorial objects.

METHODS:

This course is primarily lecture-based. Evaluation includes quizzes, tests, and a final exam.

PRIOR LEARNING ASSESSMENT RECOGNITION (PLAR):

Credit can be awarded for this course through PLAR (Please check :) ☒ Yes ☐ No

METHODS OF OBTAINING PLAR:

Course challenge.

TEXTBOOKS, REFERENCES, MATERIALS:

[Textbook selection varies by instructor. An example of texts for this course might be:]

The textbook is chosen by a departmental curriculum committee. Recent text used:
Straight, H. Joseph. 1993. *Combinatorics: An Invitation.* Brooks Cole.

SUPPLIES / MATERIALS:

STUDENT EVALUATION:

[An example of student evaluation for this course might be:]

The weighting of the various components may vary from instructor to instructor and from year to year, although there must be at least two midterms, and the comprehensive final exam must be worth from 30% to 50% of the final grade. A student must obtain at least 40% on the final exam in order to pass this course.

Quizzes	10%
Assignments	10%
Tests (2)	40%
Final Exam	40%

COURSE CONTENT:

[Course content varies by instructor. An example of course content might be:]

Topics covered include all items in 1, 2 and 3 below and selected items from 4, 5 and 6 below.

1. Concepts of combinatorics and graph theory – combinations, permutations, partitions, paths and cycles.
2. Computer representation of combinatorial objects – representations of integers, sets, graphs, etc.
3. Complexity of combinatorial computations – computational efficiency, polynomial time algorithms and satisfiability problems.
4. Basic techniques – search in trees, generating functions, permutations and partitions of a set, sorting problems.
5. Shortest paths and flow problems in networks – max-flow min-cut theorem, the labeling problem, Dijkstra's algorithm, Floyd-Marshall method and their applications.
6. Other algorithms – graph colouring, Hamilton path generation, the traveling salesman problem, matching problem.