UNIVERSITY COLLEGE of the FRASER VALLEY

COURSE IMPLEMENTATION DATE: September 1994
COURSE REVISED IMPLEMENTATION DATE:
COURSE TO BE REVIEWED: September 1998
(Four years after implementation date) (MONTH YEAR 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: **MATHEMATICS**

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

Applied Discrete Mathematics
COURSE DESCRIPTIVE TITLE

CALENDAR DESCRIPTION:

An introduction to discrete modeling, generation of combinatorial objects, applications to scheduling, applications of graphs.

PREREQUISITES: **Math 243, Knowledge of a computing language acceptable to the department.**
COREQUISITES: **None**

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: _____
EXPECTED FREQUENCY OF COURSE OFFERINGS: _____

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 Curriculum Committee
Chairperson: *(Curriculum Committee)*
Department Head: Barry Garner
Dean: J.D. Tunstall
PAC Approval in Principle Date:
PAC Final Approval Date: October 27, 1993

LEARNING OBJECTIVES / GOALS / OUTCOMES / LEARNING OUTCOMES:

This course is a survey of some combinatorial aspects of computing. Students will study algorithms for enumeration, optimization and other discrete problems, and their implementation on a computer. Issues of complexity will also be discussed. Some of the assignments will require use of computer resources.

METHODS:

PRIOR LEARNING ASSESSMENT RECOGNITION (PLAR):

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

METHODS OF OBTAINING PLAR:

TEXTBOOKS, REFERENCES, MATERIALS:

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

TBA

SUPPLIES / MATERIALS:

STUDENT EVALUATION:

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

Students will be evaluated on the basis of two or three in-class exams (approx. 40%), a final exam (approx. 40%) and assignments (approx. 20%). Some of the assignments will require computer resources.

COURSE CONTENT:

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

Topics will include the following items in sections 1, 2 and 3, and selected items from sections 4, 5 and 6.

Concepts of combinatorics and graph theory. (Combinations, permutations, partitions, networks, paths, cycles.) Enumeration of these objects.

Computer representation of combinatorial objects. (Representations of integers, sets, graphs, networks etc.)

Complexity of combinatorial computations. (Computational efficiency, polynomial-time algorithms, recognition problems, the satisfiability problem, Cook's theorem, Karp's reductions.)

Basic Techniques. (Searching in trees, backtracking, generation of combinations, permutations. Partitions of a set. Sieving processes and isomorph rejection. Enumeration. Sorting problems.)

Shortest paths and flows in networks. (Max-flow min-cut theorem. The labeling problem. Methods of finding shortest paths, the Belman-Ford algorithm, Dijkstra's algorithm, the Floyd-Marshall method. Minimal cost flow problems, the Hitchcock problem; applications.)

Other algorithms. (Graph colouring, backtracking, impasse detection. Hamilton path generation. The traveling salesman problem. Determination of connectivity, components, and spanning trees. Matching problems).