UNIVERSITY COLLEGE of the FRASER VALLEY

PAC FORM #2 (Page 1)

COURSE IMPLEMENTATION DATE:	Jan, 1994
COURSE REVISED IMPLEMENTATION DATE:	Sept, 2005
COURSE TO BE REVIEWED:	Sept, 2009
(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: **Science, Health & Human Services / Mathematics & Statistics**

COURSE NAME/NUMBER	FORMER COURSE NUMBER	UCFV CREDITS
MATH 360		**3**

COURSE DESCRIPTIVE TITLE: **Operations Research I**

CALENDAR DESCRIPTION:

This course is concerned with the application of mathematical models to problems arising in industry. Operations research was developed during and just after the last world war, and has had amazing success in enabling organizations to be more effective and efficient. The topics covered include: a brief review of linear programming; dynamic and integer programming, scheduling; nonlinear programming, optimization with and without constraints; network models and applications; and PERT and CPM.

PREREQUISITES: **MATH 211, MATH 221. MATH 308 recommended and will be required fall 2006**
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: April 28, 2004

LEARNING OBJECTIVES / GOALS / OUTCOMES / LEARNING OUTCOMES:

This course begins with a review of the simplex method and then proceeds to advanced topics of the simplex algorithm and applications. This course gives students the opportunity to apply their knowledge of linear algebra and to develop their ability to model real-world problems mathematically. Emphasis will be placed on understanding the mathematics behind the algorithms, their strengths, and their limitations. However, the main goal of the course is to give students the skill and techniques needed to apply powerful mathematical algorithms to solve real-world problems.
Students will use current linear programming software.

On completion of the course, the student will be able to:

1. Apply the revised simplex algorithm and the product form of the inverse.
2. Formulate Karmarkar's method for solving linear programming problems.
3. Formulate word problems such as network problems and apply the simplex network algorithm by hand.
4. Apply the branch-and-bound and enumeration methods to complex problems.
5. Formulate and solve two-person zero-sum games and be familiar with the properties of non-zero two-person games and n-person games.
6. Apply the basic methods of nonlinear programming such as the method of steepest ascent, the Kuhn-Tucker conditions, quadratic and separable programming.
7. Formulate a dynamic programming problem.

METHODS:

This course is primarily lecture-based.

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 text is chosen by a departmental curriculum committee. Recent text used:
Winston, W. and Venkataramanan, M. 2003. *Introduction to Mathematical Programming.* 4th edition. Thompson 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 tests, 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 the course.

An example of student evaluation for this course:

Assignments	35%
Tests (2)	30%
Final Exam	35%

COURSE CONTENT:

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

1. Review of the simplex algorithm.
2. Advanced topics of the simplex algorithm: the revised simplex method, product form of the inverse, the Danzig-Wolfe decomposition algorithm, complexity of the simplex method, Karmarkar's method.
3. Minimal cost network flows: the minimal cost network flow problem, some basic definitions and terminology from graph theory, the simplex method for network flow problems. Shortest path and maximum flow problems. PERT and CPM. The transportation problem, the Hungarian algorithm for solving the assignment problem.
4. Integer programming: formulating integer programming problems. The branch-and-bound method. The cutting plane method. The knapsack and traveling salesperson problems.
5. Introduction to game theory: two person zero-sum games. Saddle points. Pure and mixed strategies. Solving two-person zero-sum games via linear programming.
6. Dynamic programming: formulation and solution, Bellman's principle of optimality.
7. Applications of dynamic programming: scheduling, inventory control with deterministic demand.
8. Nonlinear programming: optimization without constraints, the one-dimensional search procedure, the gradient search procedure.
9. Optimization with constraints: the Karush-Kuhn-Tucker conditions, quadratic programming.
10. Separable programming: convex programming, Frank-Wolfe algorithm, non-convex programming, SUMT.
11. Applications of nonlinear programming: financial planning and operations management.