UCFV
UNIVERSITY COLLEGE of the FRASER VALLEY

OFFICIAL UNDERGRADUATE COURSE OUTLINE (page 1)

COURSE IMPLEMENTATION DATE:	September 1997
COURSE REVISED IMPLEMENTATION DATE:	September 2008
COURSE TO BE REVIEWED:	February 2012
(four years after UPAC approval)	*(month, year)*

OFFICIAL UNDERGRADUATE 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 – see course syllabus available from instructor

CIS 280	CIS	4
COURSE NAME/NUMBER	FACULTY/DEPARTMENT	UCFV CREDITS

Client/Server Programming
COURSE DESCRIPTIVE TITLE

CALENDAR DESCRIPTION:

This course examines the development of connectivity solutions for Client/Server systems. Topics include servers versus clients, SQL-based database systems, transactions, middleware, and communication between clients and servers. A major team-based programming project will be developed and presented by the students at the end of term.

PREREQUISITES: CIS 145 and CIS 180 with C+ or better, or COMP 155 with C+ or better
Note: As of September 2009, prerequisites will change to the following: COMP 155 with C+ or better

COREQUISITES:

PRE or COREQUISITES: CIS 230, CIS 270

SYNONYMOUS COURSE(S):

(a) Replaces: **CIS 240**

(b) Cross-listed with:

(c) Cannot take: **CIS 240** for further credit.

SERVICE COURSE TO: *(department/program)*

TOTAL HOURS PER TERM: **60**

STRUCTURE OF HOURS:

Lectures:	**45**	Hrs
Seminar:		Hrs
Laboratory:	**15**	Hrs
Field experience:		Hrs
Student directed learning:		Hrs
Other (specify):		Hrs

TRAINING DAY-BASED INSTRUCTION:
Length of course:
Hours per day:

OTHER:
Maximum enrolment: **35**
Expected frequency of course offerings: Once a year
(every semester, annually, every other year, etc.)

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

Course designer(s):	**Gary Ridsdale**		
Department Head:	**Ora Steyn**	Date approved:	**January 18, 2008**
Supporting area consultation (UPACA1)		Date of meeting:	**February 1, 2008**
Curriculum Committee chair:	**Edward Lo**	Date approved:	**January 17, 2008**
Dean/Associate VP:	**Ian McAskill**	Date approved:	**January 23, 2008**
Undergraduate Program Advisory Committee (UPAC) approval		Date of meeting:	**February 29, 2008**

LEARNING OUTCOMES:

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

- Identify how real-world databases are constructed and accessed and how transactions can be generated using Java programming.
- Evaluate the management and dynamics of group projects.

METHODS: *(Guest lecturers, presentations, online instruction, field trips, etc.)*

CIS 280 is a hands-on lab course where practical experience with visual tools and database techniques are developed.

METHODS OF OBTAINING PRIOR LEARNING ASSESSMENT RECOGNITION (PLAR):

☒ Examination(s) ☒ Portfolio assessment ☒ Interview(s)

☐ Other (specify):

☐ PLAR cannot be awarded for this course for the following reason(s):

TEXTBOOKS, REFERENCES, MATERIALS:

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

None

SUPPLIES / MATERIALS:

None

STUDENT EVALUATION:

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

Written or programming assignment every other lab period
- worth 15% of final grade
- each due two weeks after it is given

Midterm in November
- worth 10%

Final in usual Finals week
- worth 25%

Programming project
- due at the end of term
- group-oriented
- worth 50% including code, documentation, and presentation

Peer evaluation may be employed in grading a group-based term project.

COURSE CONTENT:

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

Topics include:
- Servers and clients
- How real-world databases are constructed and accessed
- Transactions
- Middleware
- Java programming for networks
- The SQL language: the *lingua franca* of client-server systems
- Communication between clients and servers
- Distributed objects
- Client/Server and the Internet