



COURSE IMPLEMENTATION DATE: January 2004
 COURSE REVISED IMPLEMENTATION DATE: September 2010
 COURSE TO BE REVIEWED: January 2014
(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 371	Computer Information Systems	3
COURSE NAME/NUMBER	FACULTY/DEPARTMENT	UCFV CREDITS
Object Oriented Modeling and Design		
COURSE DESCRIPTIVE TITLE		

CALENDAR DESCRIPTION:

Using object-oriented techniques, students will learn how to design and implement high quality computer systems. Emphasis is placed on creating a requirements model and using design principles to create a working system (user interface, application, and database). The unified modeling language (UML) is used extensively throughout this course.

PREREQUISITES: CIS 270. Acceptance to the CIS degree program. (Students accepted to a CIS or Computing Science minor may register with permission of the department.)

As of September 2011, prerequisites will change to the following:
 Acceptance to the CIS degree program; CIS 230, CIS 270, and COMP 251. (Students accepted to a CIS or Computing Science minor may register with permission of the department.)

COREQUISITES:
 PRE or COREQUISITES:

SYNONYMOUS COURSE(S):	SERVICE COURSE TO: <i>(department/program)</i>
(a) Replaces: _____	_____
(b) Cross-listed with: _____	_____
(c) Cannot take: _____ for further credit.	_____

TOTAL HOURS PER TERM: <u>45</u>	TRAINING DAY-BASED INSTRUCTION:
STRUCTURE OF HOURS:	Length of course: _____
Lectures: <u>45</u> Hrs	Hours per day: _____
Seminar: _____ Hrs	
Laboratory: _____ Hrs	
Field experience: _____ Hrs	OTHER:
Student directed learning: _____ Hrs	Maximum enrolment: <u>35</u>
Other (specify): _____ Hrs	Expected frequency of course offerings: <u>Once a year</u> <i>(every semester, annually, every other year, etc.)</i>

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): <u>Paul Rushton</u>	Date approved: <u>October 16, 2009</u>
Department Head: <u>Ora Steyn</u>	Date of meeting: <u>November 6, 2009</u>
Supporting area consultation (Pre-UPAC)	Date approved: <u>October 8, 2009</u>
Curriculum Committee chair: <u>Edward Lo</u>	Date approved: <u>January 18, 2010</u>
Dean/Associate VP: <u>Dan Ryan</u>	Date of meeting: <u>January 29, 2010</u>
Undergraduate Program Advisory Committee (UPAC) approval	

LEARNING OUTCOMES:

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

- Explain what constitutes a good requirement
- Use UML to model system requirements
- Use UML to model system design
- Apply design principles
- Translate requirements and design into good quality programs

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

- Lectures
- Review of case studies
- Programming assignments
- Formal examinations
- Student presentations

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:]

Larman, Craig. Applying UML and Patterns (3rd edition), Prentice Hall, 2005

SUPPLIES / MATERIALS:

None

STUDENT EVALUATION:

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

Assignments – 70%

Examinations – 30%

COURSE CONTENT:

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

- Object-oriented review
- Developing requirements
- Modeling with classes
- User centric design
- Modeling interactions and behaviours
- Architecting and designing software
- Testing and inspecting to ensure high quality