

COURSE IMPLEMENTATION DATE: []
 COURSE REVISED IMPLEMENTATION DATE: [September 1997]
 COURSE TO BE REVIEWED: [September 2001]
 (Four years after implementation date)

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: Computer Information Systems

COMP 455 FORMER COURSE NUMBER 3
UCFV CREDITS

Advanced Topics

COURSE DESCRIPTIVE TITLE

CALENDAR DESCRIPTION:

This course will discuss current topics in programming. The purpose of this course is to bring prospective BCIS graduates about to enter the job market up-to-date in the programming field.

PREREQUISITES: COMP 351
 COREQUISITES:

SYNONYMOUS COURSE(S) (a) Replaces: _____ <i>(Course #)</i> (b) Cannot take: _____ For further credit. <i>(Course #)</i>	SERVICE COURSE TO: _____ <i>(Department/Program)</i> _____ <i>(Department/Program)</i>
---	---

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	
Student Directed Learning: _____ Hrs	
Other (Specify): _____ Hrs	

MAXIMUM ENROLLMENT: 35

EXPECTED FREQUENCY OF COURSE OFFERINGS: Once per 12-18 months

WILL TRANSFER CREDIT BE REQUESTED? (lower-level courses only)	<input type="checkbox"/> Yes	<input checked="" type="checkbox"/> No
WILL TRANSFER CREDIT BE REQUESTED? (upper-level requested by department)	<input type="checkbox"/> Yes	<input checked="" type="checkbox"/> No
TRANSFER CREDIT EXISTS IN BCCAT TRANSFER GUIDE:	<input type="checkbox"/> Yes	<input checked="" type="checkbox"/> No

AUTHORIZATION SIGNATURES:

Course Designer(s): Gary Ridsdale Chairperson: _____
(Curriculum Committee)

Department Head: Paul Franklin Dean: Karen Evans

PAC Approval in Principle Date: _____ PAC Final Approval Date: January 29, 1997

COMP 455 Distributed Programming
COURSE NAME/NUMBER

LEARNING OBJECTIVES / GOALS / OUTCOMES / LEARNING OUTCOMES:

Students will study and gain hands-on experience with the technology of programming in a distributed computing environment.

The main concept of this course is provide third and fourth year degree students with advanced technical knowledge and experience with the design and deployment of complex distributed software. Distributed systems can be considered to be the place where the two major streams of the UCFV CIS program, programming and networking, meet. This course will focus on managing distributed *computation*, as distinct from managing distributed *data*.

METHODS:

COMP 455 is a lab-based course based on case studies and extensive development practice. The course will be delivered in lecture-lab format with numerous demonstrations. The lab portion will give students and the instructor the ability to view and interact with current projects.

PRIOR LEARNING ASSESSMENT RECOGNITION (PLAR):

Credit can be awarded for this course through PLAR

Yes

No

METHODS OF OBTAINING PLAR:

TEXTBOOKS, REFERENCES, MATERIALS:

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

Advanced Java Networking

SUPPLIES / MATERIALS:

2 formatted diskettes, 2 large envelopes

STUDENT EVALUATION:

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

Grading will be based on the students successfully completing a number of short assignments and a large programming project.

Peer evaluation will be employed in grading the group-based term project.

COURSE CONTENT:

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

Introduction

Models of distributed computation: Parallel Threads, Queue-based Message Passing.

Remote Procedure Call (RPC).

Microsoft's Distributed Component Object Model (DCOM).

OMG's Common Object Request Broker Architecture (CORBA).

JavaSoft's Java/Remote Method Invocation (Java/RMI).

Remote Method Invocation.

Role of RMI Registry.

Comparisons of CORBA and DCE.

Detailed study of programming examples.

Request brokering: Java RMI vs. CORBA Visibroker vs. Microsoft DCE.

Web objects and computation in the large.

Design and implementation of software agents.

Security Concerns