

COURSE DESCRIPTION

Dept., Number	CS447	Course Title	Distributed Objects
Semester hours	3	Course Coordinator	Dr. Shangping Ren, Assistant Professor

Current Catalog Description

This course provides an introduction to the architecture, analysis, design, and implementation of distributed, multi-tier applications using distributed object technology. The course focuses on the services and facilities provided by an Object Request Broker (ORB). Students will use a commercially available ORB and Database Management System to develop distributed object applications. Prerequisite: CS 445. (3-0-3) (T) (C)

Textbook

Gerald Brose, Keith Duddy, and Andreas Vogel, "Java Programming with CORBA, Third Edition," John Wiley & Sons, (January 2001) ISBN: 0-471-37681-7

Wolfgang Emmerich, "Engineering Distributed Objects" John Wiley & Sons, (Reprinted January 2004) ISBN: 0-471-98657-7

References

See <http://www.cs.iit.edu/~cs447>

Course Outcomes

Students should be able to:

- Understand the basic concept of distributed systems and distributed objects
- Understand the principles of Object-Oriented Middleware and common design problems for distributed systems
- Understand advantages and disadvantages of various multi-tier software architectures
- Use IDL to define application interfaces
- Use business objects to construct software applications
- Understand functions of an Object Request Broker (ORB), common distributed services, common distributed messaging styles, multiple mechanisms for providing object persistence used in distributed applications
- Understand and be able to use iterative, use case driven methodology in component-based software development
- Implement a distributed, multi-tier application using distributed object technology
- Acquire software development team-working skills using a use case driven, architecture-centric, iterative software development process

Relationship between Course Outcomes and Program Outcomes

The following Program Outcomes are supported by the above Course Outcomes:

- b. An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution
- c. An ability to design, implement and evaluate a computer-based system, process, component, or program to meet desired needs
- h. Recognition of the need for, and an ability to engage in, continuing professional development
- i. An ability to use current techniques, skills, and tools necessary for computing practices.
- k. An ability to apply design and development principles in the construction of software systems of varying complexity.

Prerequisites by Topic

- Fundamental aspects of the object-oriented model: abstraction, encapsulation, inheritance, and aggregation.
- Fundamental aspects of developing object-oriented software: requirements, analysis, design, implementation, testing, and deployment.
- Basic object-oriented design patterns: Singleton, Proxy, Abstract Factory, and Strategy.
- Experience writing object-oriented software using a common object-oriented programming language.
- Experience using a relational database management system.

Major Topics Covered in the Course

1. Course Introduction	1.0 hour
2. Software Architectures, and Business Object Architecture	4.0 hours
3. OMG Object Management Architecture, and CORBA Overview	1.5 hours
4. Interface Definition Languages, and Distributed Programming	4.5 hours
5. Project Overview	1.0 hour
6. Business Object, and Use Case Modeling	3.0 hours
7. Common Distributed Services	1.5 hours
8. Directory Services	1.5 hours
9. Persistence	3.0 hours
10. Midterm Exam	3.0 hours
11. Object to Relational Mapping, and Persistence Frameworks	3.0 hours
12. Event, Notification, and Messaging Services	3.0 hours
13. Object Database Management Systems	3.0 hours
14. Transaction Service	3.0 hours
15. Object Activation	3.0 hours
16. Application Servers, and Component Frameworks	3.0 hours

17. Future Trends	3.0 hours
	45 hours

Assessment Plan for the Course

End of every semester Course Objective Assessments by CS department. End of semester Course Evaluations by IIT. Reviewed every Spring semester by CS Undergraduate Studies Committee for possible updates in the following Fall. Once every 4-5 years a detailed review of all materials for the course is made by the CS Undergraduate Studies Committee.

How Data in the Course is Used to Assess Program Outcomes (unless adequately covered already in the assessment discussion under Criterion 4)

See the assessment discussion under Criterion 4

For a computer science program

Estimate Curriculum Category Content (Semester hours)

Area	Core	Advanced	Area	Core	Advanced
Algorithms			Software design		
Data structures			Concepts of programming languages		3