The University of Georgia

Department of Computer Science

Course Information Sheet

CSCI 4780

Distributed Computing Systems

Brief Course Description
(50-words or less)

The fundamental concepts in distributed computing and the practical techniques for building distributed systems. Topics include distributed computing models, naming, synchronization, replication and consistency, fault tolerance, and security. Widely deployed distributed systems are used as case studies. Students design, implement, and analyze prototype systems.

Extended Course Description / Comments

This course is targeted for undergraduates in their junior/senior years.

Pre-Requisites and/or Co-Requisites

Data structures (CSCI 2720) is a pre-requisite and Operating Systems (CSCI 4730) or Computer Networks (CSCI 4760)

Approved Textbooks
(if more than one listed, the textbook used is up to the instructor's discretion)

Author(s): Andrew S. Tanenbaum and Maarten Van Steen
Title: Distributed Systems: Principles and Paradigms
Edition: Second Edition
ISBN-13: 0-13-239227-5

Specific Learning Outcomes (Performance Indicators)

This course presents the fundamental concepts in distributed computing systems. At the end of the semester, all students will be able to do the following:

1. Define, identify and distinguish various types of transparencies.
2. Develop software prototypes applying variety of distributed system architecture s.
3. Define, recognize and distinguish various types of communication (synchronous, asynchronous, persistent, transient).
4. Outline the steps of a remote procedure call (RPC)
5. Develop a multi-threaded software that applies thread synchronization functionality (locks and barriers).
6. Outline the steps involved in resolving flat/structured names using recursive and iterative approaches.
7. Hand-simulate operations of logical clocks (Lamport and Vector clocks).
8. Define and distinguish various types of data consistency models
9. Develop software that applies the caching principle to world wide web content.
10. Outline the steps of two-phase and three-phase distributed commit protocols.

Relationship Between Student Outcomes and Learning Outcomes

		Student Outcomes										
		a	b	c	d	e	f	g	h	i	j	k
Learning Outcomes	1	•								•		
	2	•	•	•						•		•
	3	•								•		
	4	•										
	5	•		•	•					•		•
	6	•										
	7	•									•	
	8	•									•	
	9	•	•	•	•							•
	10	•									•	

Student Outcomes

a. An ability to apply knowledge of computing and mathematics appropriate to the discipline.
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.
d. An ability to function effectively on teams to accomplish a common goal.
e. An understanding of professional, ethical, legal, security and social issues and responsibilities.
f. An ability to communicate effectively with a range of audiences.
g. An ability to analyze the local and global impact of computing on individuals, organizations, and society.
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 practice.
j. An ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices.
k. An ability to apply design and development principles in the construction of software systems of varying complexity.

Major Topics Covered
(Approximate Course Hours)

3 credit hours = 37.5 contact hours
4 credit hours = 50 contact hours

Note: Exams count as a major topic covered

Goal and Types of Distributed Systems (4-hours)
Distributed System Architectures (4-hours)
Threads and processes in distributed systems (5-hours)
Communication models (2-hours)
Remote procedure calls, sockets and MPI (5-hours)
Naming and name resolution (5.5-hours)
Physical and Logical Clocks (4-hours)
Mutual Exclusion and Leader Election (2-hours)
Data Centric Consistency Models (2-hours)
User Centric Consistency Models (2-hours)
Replica Management (1-hr)
Reliability in Client-Server Architectures (2-hours)
Distributed Commit Protocols (2-hours)
Authentication and Access Control in Distributed Systems (4-hours)
Distributed File Systems (3.5-hours)
World Wide Web-based Systems (4-hours)

Assessment Plan for this Course

Each time this course is offered, the class is initially informed of the Course Outcomes listed in this document, and they are included in the syllabus. At the end of the semester, an anonymous survey is administered to the class where each student is asked to rate how well the outcome was achieved. The choices provided use a 5-point Likert scale containing the following options: Strongly agree, Agree, Neither agree or disagree, disagree, and strongly disagree. The results of the anonymous survey are tabulated and results returned to the instructor of the course.

The course instructor takes the results of the survey, combined with sample student responses to homework and final exam questions corresponding to course outcomes, and reports these results to the ABET committee. If necessary, the instructor also writes a recommendation to the ABET committee for better achieving the course outcomes the next time the course is offered.

How Data is Used to Assess Program Outcomes

Each course Learning Outcome, listed above, directly supports one or more of the Student Outcomes, as is listed in "Relationships between Learning Outcomes and Student Outcomes".

Course Master

Dr. Lakshmish Ramaswamy

Course History

05/2007	Course Information Uploaded to CAPA
02/2012	Course Information Sheet Created