

ECE 335
Integrated Software Systems
Spring 2008

Description: This course covers the foundation of core computer engineering topics that bridge the gap between the basic computer organization, programming and the 4xx-level courses, such as Operating Systems, Networks, Graphics, Software Engineering.

This course will focus on engineering design, implementation, and performance evaluation of software systems.

Topics include two sub-areas:

- i) software and underlying architectures, and
- ii) software and integrated systems.

Main objectives:

- Students will learn basic engineering design concepts of integrated system software components, learn how to apply them to a large-scale engineering environment, and how to evaluate performance trade-offs.
- Students will gain the preview of important complex system engineering issues concerning manipulation and management of distributed entities, resource allocation, information servers, naming and communication.
- Students will learn how to evaluate scalability, reliability, security, concurrency, efficiency, and other engineering design features.
- This knowledge will be used as a pre-requisite for higher-level classes.

Table I: Objectives, Implementation, and Assessment

Objectives		Implementation	Assessment	A	B	C	D	E	F	G	H	I	J	K
O ₁	Understand intermediate hardware and software structures, their relevance to specific software applications	3 hrs. lecture in 1 st week	HW 1, Exam I	X	X	X		X						
O ₂	Understand ways to evaluate performance of integrated systems. Understand typical approaches to optimize performance. Understand trade-offs associated with various optimization techniques	6 hrs. lecture in 2 nd and 3 rd weeks, and 1.5 hrs in week 6	HW 2, Exam I	X	X	X		X						
O ₃	Understand impacts of hardware architecture (CPU, memory, system I/O) on performance. ReUnderstand trade-offs associated with architecture and performance optimization	12 hrs. lecture in weeks 4-7	HW 3,4 Exam I	X	X	X		X						
O ₄	Understand system integration approaches, typical applications and different programming languages. World Wide Web integrated interfaces.	9 hrs. lecture in weeks 8-10	HW 5,6 Project. 1 Exam II	X	X	X	X	X						
O ₅	Understand different Web services and architectures. Information servers and communication, network programming	9 hrs. lecture in weeks 10-12	HW 7,8 Exam II	X	X	X		X						
O ₆	Gain experience implementing intermediate-scale integrated systems, be able to evaluate performance and other parameters (reliability, scalability, security, etc) of designed systems	Remaining lectures in weeks 13-16, programming project assignments	Proj. 2, Exam II	X	X	X	X	X						

**Table II: Expectation and Assessment Outcome
Spring 2008, Dr. Olga Lavrova**

General expectations:

Homeworks:

These assignments involve two components:

1. sharpen/practice/improve skills and understanding of experiment and system design
2. simple and intermediate programming exercises aimed at preparing students for programming projects (e.g., making sure they are able to evaluate performance using different metrics, be able to implement required networking operations and using different programming languages, etc.).

For (1), expect that all of the students will be able to answer 100% of the problems assigned.

For (2), Expect 75% of the students to score 80% or better on the problems assigned

Programming Projects:

Project 1 – Requires analysis of a problem and different ways to solve it, identifying the most efficient for a given hardware architecture and resource allocation

Project 2 – More advanced web services or network programming project, tied to real-life programming application

Exams: Expect 75% of the students to score 80% or better on the exams.

	Objectives	Assessment	Outcome measurement
O ₁	Understand intermediate hardware and software structures, their relevance to specific software applications	HW 1, Exam I	√ Homework #1: 88% ≥80% √ Exam 1: 77% ≥80%
O ₂	Understand ways to evaluate performance of integrated systems. Understand typical approaches to optimize performance. Understand trade-offs associated with various optimization techniques	HW 2, Exam I	√ Homework #2: 77% ≥80% √ Exam 1: 77% ≥80%
O ₃	Understand impacts of hardware architecture (CPU, memory, system I/O) on performance. ReUnderstand trade-offs associated with architecture and performance optimization	HW 3,4 Exam I	√ Δ Homeworks #3 and 4: 77% ≥80% √ Exam 1: 77% ≥80%
O ₄	Understand system integration approaches, typical applications and different programming languages. World Wide Web integrated interfaces.	HW 5,6 Project. 1 Exam II	√ Homeworks #5 and 6: 77% ≥80% √ Project 1: 77% ≥80% √ Exam 2: 77% ≥80%
O ₅	Understand different Web services and architectures. Information servers and communication, network programming	HW 7,8 Project 2 Exam II	√ Homeworks #7 and 8: 77% ≥80% √ Project 2: 77% ≥80% √ Exam 2: 77% ≥80%

O ₆	Gain experience implementing intermediate-scale integrated systems, be able to evaluate performance and other parameters (reliability, scalability, security, etc) of designed systems	Project 2, Exams II	√ Project 2: 77% ≥80% √ Exam 2: 77% ≥80%
----------------	--	---------------------	---

√ means passing expectation
 Δ means needing improvement

Homework #3. Students had difficulty evaluating results due to differences in students' personal computers configuration. This was addressed by suggesting students use ECE departments' infrastructure and computers that were available to students.

Required Textbook: *Computer Systems, A Programmers' Perspective*, Randal E. Bryant and David O'Hallaron, Prentice Hall, 2003, ISBN 0-13-034074-X

Additional Textbooks: *Understanding Web Services: XML, WSDL, SOAP, and UDDI*, Eric Newcomer, Addison-Wesley, ISBN: 0-201-75081-3
Java Web Services for Experienced Programmers, Prentice Hall, 2002, ISBN: 0130461342

Sample Course Schedule

Spring 2008 Schedule

Week 1 (01/21):	Hardware, Software and Application Structures - Overview
Week 2 (01/28):	Performance measurement
Week 3 (02/04):	Performance optimization
Week 4 (02/11):	Impacts of CPU architecture on programming
Week 5 (02/18):	Impacts of memory organization on programming
Week 6 (02/25):	More on optimization
Week 7 (03/03):	Runtime and system I/O
Week 8 (03/10):	System integration
Week 9 (03/24):	More on integration, Scripting Languages
Week 10 (03/31):	World Wide Web Integrated Interfaces
Week 11 (04/07):	Project 1 presentations
Week 12 (04/14):	Web service, client/server models
Week 13 (04/21):	Web service, client/server models – continued
Week 14 (04/28):	Integration with database backend engines
Week 15 (05/05):	Project 2 presentations
Week 16 (05/12):	Final Exam