University At Buffalo
COURSE OUTLINE

A. Course Title: **CSE 487/587 Information Structures**

B. Curriculum: Computer Science and Engineering Offering

   Program: BS, BA, MS and Ph.D

C. Catalog Description: 4 Credits Elective

The objective of the course is to give an overview of the use of information technology in large-scale commercial and scientific systems, with emphasis on the use of state of the art computing in realizing various services and the frameworks supporting these services. Concepts covered include: enterprise modeling, process modeling, process automation and streamlining, workflow management, messaging, persistent message queues, transaction monitoring, document exchange, application servers, service definition (web services, web services definition language: WSDL), connection and resource reservation protocols (TCP, grid computing), integration technologies and architectures (Java 2 Enterprise Edition: J2EE, eXtensible Markup Language: XML, and Globus toolkit).

D. Class Timing and Location: Tu/Th 8.00-9.20AM , Baldy 200G

E. Suggested Text: The Grid: Blueprint for a New Computing Infrastructure by I. Foster and C. Kesselman. Second Edition, Morgan Kaufmann Publishers, 2003.

Tools/Languages: Java SDK1.4, Java 2EE, XML, SOAP, Globus/Grid development environment

F. Instructor Information: **Instructor: B. Ramamurthy (Bina)**
       127 Bell Hall
       bina@cse.buffalo.edu
       http://www.cse.buffalo.edu/~bina
       716-645-3180 (108)
       Office Hours: MWF: 10-10.50AM

G. Program Competencies: ACM Curriculum 2001 suggests a model curriculum. Among the topics specified in that curriculum, this course will cover:

1. NC1: Net-centric Computing: Distributed Systems: Describe emerging technologies in the net-centric computing area and assess their current capabilities, limitations, and near-term potential. (Addressed by objective 1 through 7).

2. NC5: Building Web Applications: The Role of middleware, enterprise-wide web applications: Implement a distributed system using any two distributed object frameworks and compare them with regard to performance and security issues. (Addressed by objectives 2,3,4 and 6)

3. SE2: Using APIs: Design, implement, test, and debug programs that use large-scale API packages. (Addressed by objectives 2 and 6)

4. SE3: Software Tool and environments: Programming environments, testing tools, scripting and configuration management tools. (Addressed by objective 4)

5. SE9: Component-based programming: Apply component-oriented approaches to the design of a range of software including those required for concurrency and transactions, reliable communication services, database interaction including services for remote query and database management, secure communication and access. (Addressed by objective 2)

6. SE12: Specialized System Development: Distributed Systems: Grid Systems: Outline the central technical issues associated with the implementation of specialized systems development. (Addressed by course objectives 1 through 4)

7. CN4: High Performance computing: Recognize problem areas where computational modeling enhances current research methods (such as grid computing). (Addressed by course objectives 1, 3 and 5).

H. Course Objectives: At the completion of the course, the student should be able to:

1. Understand the application context of large scale distributed systems. For example: Scientific and industrial applications, from biomedical to astrophysics.
2. Work with the current application models: component-oriented, and grid-oriented.
3. Analyze and design the infrastructure needs of a large scale distributed system. For example: computing elements, configuration, deployment details, peer-to-peer communication.
4. Work with development tools (Ex: Eclipse integrated development environment, Apache Ant, and JUnit).
5. Understand and work with open grid services architecture and infrastructure.
6. Design and implement using Globus grid computing framework.
7. Technology Objectives: Students will be able to demonstrate the ability to design, implement, and deploy distributed systems based on Java technology and Grid Technology.

I. Assessment of Student Learning:

1. A mid-semester exam will be given during the semester and a three-hour final exam during the exam week scheduled by the university.
2. We will also use additional methods of evaluation to include: Graded programming assignments/projects, and lab exercises.

J. Learning Resources and other Support: It is required that all students attend the recitation that will be conducted by a Graduate Teaching Assistant (GTA). The instructor for the courses and the GTA will have each 3 hours of open office hours each per week to help with any question related to the course. Students are encouraged to use the open hours of Computer Science Lab at 338 Bell Hall and the Grad lab. Each student will be allocated individual project space to install the appropriate software needed for the labs.

K. Tentative Schedule:

Week of	Topics Covered	Reading Material
1/12	General Introduction; Course Outline; Cleint/server systems.	Firstday handout
1/19	Distributed System using Java RMI. Enterprise Computing: technology landscape.	Class notes: Will be posted
1/26	Project 1 discussion; Setting enterprise integration design objectives. Recitation: Distributed system using servlets.	java.sun.com/j2ee
2/2	Designing the enterpirse architecture; establishing the enterpise infrastructure; JDBC and using Oracle server	Department web page
2/9	Java 2 Enterprise Edition (J2EE). Techologies, enterprise application model, design and implementation of a sample application.	java.sun.com/j2ee
2/16	Grids in context: Fundamantal of a grid	Ch. 1
2/23	The scientific grid and the industrial grid	Ch. 2 and Ch.3
3/1	Project 2 discussion: Basic Grid Application; Application level tools and runtime systems.	Project 2 handout;
3/8	Globus Toolkit discussion: Open Grid Services Architecture and Infrastructure	www.globus.org; Ch. 4
3/15	Spring Break	
3/22	Grid architecture: Resource and service management	Ch. 17, 18
3/29	Building grid clients and services; Project 3 discussion	Ch. 19-21
4/3	Globus toolkit usage. Project 3 discussion: Grid based enterprise resource management	Project 3 handout; www.globus.org
4/10	Grid applications: Selected applications from Ch. 5-16.	
4/17	Data access, integration and management	Ch. 22
4/24	Review for the final exam.	

Date	Item Due
2/24	Project 1
3/31	Project 2
4/26	Project 3
2/26	Exam 1
Finals Week	Final Exam

Lab (Project) Topics

Note: Each lab will involve complete installation of all the necessary toolkits, software packages and servers by each student (or group of students) in their workspace. Students will also write a detailed technical report on the project they implement.

Lab 1: Design, implement and deploy a web application with component-oriented middleware
4 weeks

Lab 2: Design, implement and deploy a basic grid-oriented application.
4 weeks

Lab 3: Design and implement a sophisticated applications with groups within the class interacting.   4 weeks

L. Grading Policy:

Grades will consist of the following components:

Projects (3 projects)   : 55%

Midterm Exam   : 20%

Final Exam   : 20%

Class Participation   : 5%

Grade Range	Letter Grade
95 - 100	A
90 - 94.99	A-
85 - 89.99	B+
80 - 84.99	B
75 - 79.99	B-

70 - 74.99	C+
65 - 69.99	C
60 - 64.99	C-
55 - 59.99	D+
50 - 54.99	D
0 - 49.99	F

The *Minimal* point distribution guideline will be as above. We reserve the right to alter component weighting or provide a "curve" on an assignment as warranted. In order to pass this course you must have passing average in the Exam component of the course. All assignments will be graded and returned in a timely manner. When an assignment is returned, you will have a period of one week to contest any portion of the grade. Grading conflicts will be first resolved with the TA who graded your assignment. If the conflict cannot be resolved, the instructor will mediate the dispute. When contesting a grade, you must be able to demonstrate how your particular solution is correct. Also, when contesting a grade, the instructor or TA reserves the right to re-evaluate the entire exam, not just the question in dispute.

Projects

Projects constitute a major component of the course. Students will apply the concepts studied during the lecture in three group projects: an introductory project in enterprise systems, and two large projects in grid computing.

You will be given approximately four weeks to complete each project. *Do not be lulled into a safe sense of security thinking you have a lot of time to implement each project!* Much of your project development time will be spent in the design phase of your code. When implementing your solution, you should plan on using an *incremental development* path. You should plan your project in achievable stages such that you can get parts of your solution working a little at a time. This will help maximize partial credit during grading. Late assignments will be penalized at a rate of 25% of the achieved grade for each day overdue. The penalty will be assessed from the due date and time indicated on the assignment.

Exams

There will be a midterm exam that will be administered and graded before the course resign date. Midterm material will cover all lecture and reading assignments before the exam, as well as

concepts from homework and lab assignments. The final exam is comprehensive, covering all lecture, lab, and homework areas. Make-up exams are not administered! If you miss the midterm exam or final, you will be assigned a grade of 0 points for that component.

Lecture and Recitation Attendance

Attendance is required for all lectures and assigned recitations. You are responsible for all materials presented in lecture and recitation, as well as handouts and/or other supplemental material. I do not give **incomplete**s in the course, unless under the most dire of circumstances. By definition, an incomplete is warranted if the student is capable of completing the course satisfactorily, but some traumatic event has interfered with their capability to finish within the timeframe of the semester.

M. Academic Integrity:

UB's definition of Academic Dishonesty in part is, "Students are responsible for the honest completion and representation of their work". You should also read the departmental academic honesty policy located at http://www.cse.buffalo.edu/academics-academic_integrity.shtml.

You must abide by the UB Academic Integrity policy at all times. Remember that items taken from the web are also covered by the academic honesty policy. If you are caught violating the academic integrity policy, you will minimally receive a ZERO in the course. We will also place the incident in your permanent record. If it is your second violation, we will recommend to the Undergraduate/Graduate committee Chair that formal proceedings be filed against you, which would mean either you could be expelled, or your degree progress will be terminated within the Computer Science and Engineering department.

N. Help:

When asking questions, please try and talk with the TA first. He has probably covered the question many times with other students. Take advantage of my office hours and theirs. We have about 6 hours amongst us. Attend the recitations regularly.

If you have special needs due to a **disability,** and are registered with the Office of Disability Services, we need to know as soon as possible! Do not assume that we have received the paperwork! (Although it is your responsibility to make sure we receive the paperwork as soon as possible from Disability Services).