COURSE DESCRIPTION:

An application server is a rich, highly portable software that runs on a middle tier and handles all application operations between browser-based pervasive devices and back-end databases and business applications. Application servers provide a platform independent programming interface for developing portable Java applications. Application servers also facilitate the integration of legacy applications via on-the-fly transformation of XML-formatted data, and support a wide variety of XML-enabled client channels that include traditional web clients and a growing set of smart devices. As emerging standards such as SOAP enable a new generation of "web services" that allow systems to make remote procedure calls to other systems over the Internet, application servers are setting the stage as modern platforms for Web Service platforms initiatives, application server appliances, Web services and wireless applications.

This course concentrates on architecting, designing, and developing persistent software application using application server technology. Throughout the course, students are exposed to the evolution of application server architectures that started in the mid 1990s, and experiment with corresponding approaches based on traditional client-server technology, CGI frameworks, page-based extended HTML environments, distributed object computing platforms, object management architectures, component-based computing environments, and web services platforms. The course conveys the necessary skills to select the proper application server architecture based on project requirements and scale. The course also explains how to integrate an application server into an existing Web site, as well as how to implement an application server-based Web application from the ground up. Students will learn how to configure and operate application servers for production environment taking advantage of features available in mainstream commercial frameworks such as scalability, concurrency, security, fault tolerance, auto-deployment, communications support, development environment, and monitoring
tools.

As they design and implement applications using application server technologies, students will learn how to identify application patterns that lead to the most effective use of the various services provided within application server frameworks. The design and implementation of the persistence and legacy application integration layers using related application server technology will be particularly emphasized. Case studies, provided as part of the course, focus on how medium- to large-size sites manage the complexities inherent in these endeavors. The case studies will help students get a firm understanding of how application servers work and how to best deploy complex applications in a real-world situation. Although, the course will strive to provide a complete coverage and classification of application server technology, attempts will be made whenever possible to select open source technologies for experimentation purpose. As part of the course, students will be exposed briefly to next generation reflective, multimedia- and agent-enabled application servers with support for model driven architectures.

COURSE OBJECTIVES

The objectives of the course are as follows:

1. Convey the evolution of application server architectures that started in the mid 1990s.

2. Provide a broad exposure to legacy, mainstream, and upcoming application server technology.

3. Expand the in-depth understanding of mainstream component-based application server environments based on the EJB, CORBA 3, and COM+ component models.

4. Identify and study core application server technologies.

5. Provide exposure to advanced design and development techniques using application server technology.

6. Clarify the key role of XML in application server’s support for application integration, Web Service platforms initiatives, application server appliances, and wireless applications.

7. Demonstrate the use of application server technology for business critical applications that require scalability, concurrency, security, fault tolerance, auto-deployment, communications support, development environments, and monitoring tools.
8. Provide a snapshot of upcoming technology directions that will guarantee a lasting expertise of application server technology.

**TEXTBOOKS**

Building Application Servers (Advances in Object Technology 24)  
Rick Leander  
Cambridge Univ. Press (Trd), ISBN: 0521778492 (6/00)

Professional Java 2 Enterprise Edition with BEA WebLogic Server  
Francisco Gomez, Peter Zadrozny  
Wrox Press Inc, ISBN: 1861002998 (10/00)

**PREREQUISITES**

Students enrolling in this class are expected to have taken G22.2110 (i.e., Programming Languages), and G22.2250 (i.e., Operating Systems) and their prerequisites or to have equivalent knowledge. Students are also expected to have taken a Java intermediate course, and to have basic knowledge of the Core JFC classes, and the ability to program in Java. Some exposure to the topics and techniques covered in G22.3033-02 (i.e., Programming for the World Wide Web) is a plus.

**WEB SITES**

Download Java 2 SDK from [http://www.javasoft.com/](http://www.javasoft.com/). Other sources are listed under References on the course main page. These include information available on application servers portals, application servers FAQs, Web Services platforms such as Sun's Open Net Environment (ONE), HP's NetAction/e-speak, Oracle's Dynamic Services and Portlets (Oracle 9I), Microsoft’s .NET, Novell’s Directory-Enabled Net Infrastructure Model (DENIM), IBM’s WebSphere Architecture (WSA), BEA Systems’ WebLogic, and Web Services initiatives conducted by the Apache Group, ActiveState, ObjectSpace, XMethods.com, DevelopMentor, Rogue Wave, and BowStreet. Additional sources will be provided as applicable during the course sessions.
COURSE SESSIONS

1. Legacy Application Server Technology
   - Network Programming Review
   - Traditional Client-Server Technology
   - Object-Oriented Client-Server Environments
   - CGI Frameworks
   - Ongoing Project Overview

   READINGS: Building Application Servers: Part I, Chapter 1
   Handouts posted on the course web site

2. Page-Based Application Servers (Part I)
   - ColdFusion 5.0 Environment
   - PHP 4 Environment
   - XML-Based Application Servers

   READINGS: Building Application Servers: Part I, Chapter 2
   Handouts posted on the course web site

3. Page-Based Application Servers (Part II)
   - ASP Environment (i.e., IIS with COM+, and ASP)
   - Servlets
   - JSPs and TomCat
   - XSPs

   READINGS: Building Application Servers: Part I, Chapter 2
   Handouts posted on the course web site

4. Distributed Object Computing Platforms
   - CORBA
   - RMI and RMI-IIOP
   - COM+
   - DOC Platform Interoperability
   - Web-Enabled DOC Applications

   READINGS: Building Application Servers: Part II, Ch. 1-3
   Handouts posted on the course web site

5. Object Management Architectures
   - Object Management Architectures
   - Java-Based Application Servers
   - Windows Services
6. **J2EE Component-Based Computing Environments (Part I)**

   - EJB Component Model
   - J2EE Services (JNDI, JMS, JTS, CMP/BMP/JDBC, JavaMail)
   - Security in J2EE Application Servers

7. **J2EE Component-Based Computing Environments (Part II)**

   - WebLogic
   - WebSphere
   - Open Source J2EE Environments (JBoss, Enhydra, OpenEJB)

8. **J2EE Component-Based Computing Environments (Part III)**

   - Other J2EE Application Servers
   - Inprise, iPlanet, Sybase EAServer, etc.

9. **CORBA 3 Component-Based Computing Environments**

   - CORBA 3 CCM
   - CORBA 3 Environments

10. **COM+ Component-Based Computing Environments**

    - COM/DCOM Component Model, and OLE
    - COM+ MTS services: Transactions, and Security
    - COM+ Message Queuing
11. **XML-Based Capabilities in Component-Based Environments**

- EAI Environments
- B2Bi Environments (e.g., WebMethods Framework)
- Channel Independence and Pervasive Devices

**READINGS:** Building Application Servers: Parts II, and III
Handouts/References posted on the course web site

12. **Web Services Platforms (Part I)**

- XML-RPC
- SOAP
- UDDI
- WSDL
- C# and .Net

**READINGS:** Building Application Servers: Parts II, and III
Handouts/References posted on the course web site

13. **Web Services Platforms (Part II)**

- Mainstream UDDI Registries and Browsers
- Mainstream Services Toolkits

**READINGS:** Building Application Servers: Parts II, and III
Handouts/References posted on the course web site

14. **Upcoming Application Server Technologies**

- Model Driven Architectures
- Reflective Application Servers
- Intelligent Agents
- Multimedia/Broadband Application Servers

**READINGS:** Building Application Servers: Parts II, and III
Handouts/References posted on the course web site
READINGS

Assigned readings for the course will be from the textbooks, from various web sites documentation, and from trade magazines and recommended books listed on the course web site.

ASSIGNMENTS

Homework and project assignments completion will be required. Quizzes will be administered. The final exam will be a take-home exam.

GRADING POLICY

25% Assignments
35% Projects
30% Final Exam
10% Attendance and Participation
Extra credit will be granted periodically for particularly clever or creative solutions.