Component-based Software Development  
SWE 645, Spring Semester 2006

Instructor: Dr. Nick Duan  
Email: nduan@gmu.edu  
URL: [http://ise.gmu.edu/~nduan/swe645](http://ise.gmu.edu/~nduan/swe645)  
Office Hours: Anytime electronically or 20 min before class starts on class day in  
classroom. The best way to reach me is via email.  
GTA: Vijayant Dhankhar  
GTA Email and Phone #: vdhankha@gmu.edu  
GTA Office Hours: Wednesday from 7:00 to 9:00 PM, R468, STII

Prerequisite:

- SWE 619 and SWE Foundation material (MSCS Students may substitute CS 540  
  and CS 571 for SWE 619).
- Knowledge of object-oriented programming and basics of Windows and UNIX
- Good knowledge and programming skills in Java, JDBC, and Java foundation  
classes. Knowledge of Java Servlet and JSP (SWE 642) experiences are  
desirable.

Objectives:

- Understand the concepts of component software and various design patterns and  
  component frameworks. The students will learn how component software and  
  frameworks are created and designed, and what basic issues (e.g. session,  
  persistency, security and transactions) are to be addressed in enterprise  
  component software.
- Become skillful with various technologies in Java for developing component- 
  based software frameworks and applications, including JavaBeans, Enterprise  
  Java Beans, Servlet/JSPs, JMS, distributed computing frameworks, XML-based  
  configuration, Java reflection, RMI, and JNDI.
- Become familiar with the software development life cycle and design techniques  
of large-scale applications via hands-on programming assignments.

Contents:

- Concepts of component software frameworks and development issues to be  
  addressed. Methodologies for creating customized component frameworks.
- Software design patterns (GOF, Core J2EE), decoupling via declarative  
  programming, Java reflection, serialization, and other techniques in Java.
- Component techniques: JavaBeans, event handling, naming service(JNDI)
- Container architecture: Concept of containers, J2EE Web Container and EJB  
  Container
- Packaging and deployment: Java properties, XML-based configuration, standardization of packaging directory structures for web and EJB applications
- Persistency, resource and transaction management, component instances, JDBC connectivity, and resource management in J2EE containers.
- Concepts of Java Messaging Services (JMS).
- Security and performance issues in component technologies.

Textbook:

Additional References:

Grading Policy:
- Midterm Exam 30% (open book)
- Final Exam 30% (open book)
- Project Assignment 35%
- Presentation/Innovation 5%

Project Assignment:
- Objective: to obtain hands-on experience with JavaBeans, Enterprise JavaBeans, and Servlet, and/or RMI through the development of a web-based, multi-tiered enterprise application, and to gain experience with real-world software development.
- Description: The application is an on-line flight reservation application with web front end for both the traveler and the airline administrators. The traveler should be able to search different flight, create itineraries, and book flights. The airline administrator should be able to publish and update flight information (flight number, aircraft type, departure/arrival time and locations, and prices). The actual database schema is up to each student/group to decide.
- The application shall consist of the following components:
  - A set of web pages, development in a combination of HTML, JavaScripts and/or JSPs
A set of database access objects (DAO) with JDBC connections to a backend database.
A set of Java servlets for connecting with a backend EJB server (as an EJB client), or connecting with a RMI server (as an RMI client).
A set of EJB components (including both entity beans and session beans) running on a back-end EJB server (option I), or a set of RMI objects running on a backend RMI server (option II).

The following server modules are to be used for this project:

- Sun’s Application Server (version 8.1, downloadable from Sun’s web site) that provides both EJB services and web container capabilities.
- An RMI server (bundled with J2SE 5.0).
- A database server for storing all relevant information of the application. The PointBase database bundled with the app server or any other database chosen by students may be used for development. The final demonstration may use Oracle provided by the department.

System and Deployment Requirement: The application should be developed using J2SE 5.0, and Java Servlet version 2.4, JavaServer Pages version 2.0 (using Sun’s Application server). The application server along with J2SE 5.0 will be installed on the department’s UNIX machine (hermes), available to every student. The students are encouraged to install any J2EE compliant application server on their own computers for convenient reasons. However, only support to Sun’s app server is available from the instructor and/or the TA. The final application shall be installed and demonstrated on a designated machine provided by the instructor.

Policies: The project shall be completed a group of two or three students. All students of the group will receive the same final score for the project. The deliverable consists of a design document, source code, Java docs, and deployable modules/application. Design document should describe the system architecture, components, and customized component frameworks. The deliverable as a single J2EE application by following the standard deployment structure, is to be uploaded to a designated directory on hermes provided by the instructor on or before the project due date(s).

Detailed requirement and grading scales is available on the class web site.

Additional Information and Policies:

- All class related information, including syllabus, project information, and class notes, is accessible from the instructor's web page.
- Each student shall set up his/her computer account to receive class-related emails and help information.
- Make-up exams are permitted on a case-by-case basis according to University and Department policies. Anyone who wishes to take a make-up exam must inform the instructor at least one week in advance (either in class or via email, with name and student number), and must provide the instructor with approval from legitimate sources (doctors, employer, police, etc.).
- The project due day is fixed and non-negotiable.
Honor Code Statement:

As with all GMU courses, SWE 645 is governed by the GMU Honor Code. In this course, all assignments, exams, and project submissions carry with them an implicit statement that it is the sole work of the author, unless joint work is explicitly authorized. Help may be obtained from the instructor or other students to understand the description of the problem and any technology, but the solution, particularly the design portion, must be the student's own work. If joint work is authorized, all contributing students must be listed on the submission. Any deviation from this is considered an Honor Code violation, and as a minimum, will result in failure of the submission and as a maximum, failure of the class.

Additional Recommendations:

1. This class covers a wide range of topics of component technologies in enterprise computing. It is impossible to discuss all topics listed in the syllabus in details. The students are encouraged to be pro-active both during class and after class. Additional readings and homework assignments (not checked) shall be completed in order to complete the class successfully.
2. This is a programming-intensive class. The scale and the complexity of the class project require dedication and considerable amount of time. Students who have registered for three or more classes per semester should make proper planning before taking this class.
3. This class is designed to focus on basic concepts and skills in J2EE. Although non-standard J2EE tools and frameworks are allowed in the class project (e.g. Struts, Ant, NetBeans and other IDE tools), but they will not be supported. Because of the high learning curve involved in these tools, it’s not recommended to use them in the class project without prior knowledge and experience, especially since these tools tend to hide the basic concepts from the non-standard APIs and processes.
4. Everyone is highly encouraged to ask questions. Remember, these are no dump questions, only dump answers.
<table>
<thead>
<tr>
<th>Session</th>
<th>Date</th>
<th>Topics</th>
<th>Lecture Notes</th>
<th>References</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1/24</td>
<td>Introduction to Component Software Frameworks and J2EE Platforms</td>
<td>645-Introduction</td>
<td>Roman: Ch. 1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Additional Ref.</td>
</tr>
<tr>
<td>2</td>
<td>1/31</td>
<td>JavaBeans and DAO with JDBC</td>
<td>JavaBeans, JDBC, (2)</td>
<td>Additional Ref.</td>
</tr>
<tr>
<td>3</td>
<td>2/7</td>
<td>Web Component in J2EE (Servlet, JSPs)</td>
<td>Servlets, (2)</td>
<td>Additional Ref.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Advanced Servlets, (2),</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Session, (2)</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>2/14</td>
<td>Distributed computing model, from RMI to EJB frameworks</td>
<td>RMI, (2)</td>
<td>Additional Ref.</td>
</tr>
<tr>
<td>5</td>
<td>2/21</td>
<td>Introduction to EJB (Phase I Due)</td>
<td>EJB Intro, (2)</td>
<td>Roman: Ch. 2, 3</td>
</tr>
<tr>
<td>6</td>
<td>2/28</td>
<td>Session Beans</td>
<td>Session Beans, (2)</td>
<td>Roman: Ch. 4</td>
</tr>
<tr>
<td>7</td>
<td>3/7</td>
<td>Midterm Exam (open book)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>3/21</td>
<td>JMS and Message Driven Beans</td>
<td>Message-Driven Beans, (2)</td>
<td>Roman: Ch. 9</td>
</tr>
<tr>
<td>9</td>
<td>3/28</td>
<td>J2EE Security (Phase II Due)</td>
<td>645-Security</td>
<td>Roman: Ch. 13,</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Additional Ref.</td>
</tr>
<tr>
<td>10</td>
<td>4/4</td>
<td>J2EE Transaction Management</td>
<td>Transaction, (2)</td>
<td>Roman: Ch. 12</td>
</tr>
<tr>
<td>11</td>
<td>4/11</td>
<td>Performance Issues in J2EE</td>
<td>645-Performance</td>
<td>Roman: Ch.18, 19</td>
</tr>
<tr>
<td>12</td>
<td>4/18</td>
<td>Dynamic component instantiation using Java reflection and factory</td>
<td>In Progress</td>
<td>Appendix C</td>
</tr>
<tr>
<td></td>
<td></td>
<td>pattern. Packaging and deployment</td>
<td></td>
<td>Additional Ref.</td>
</tr>
<tr>
<td>13</td>
<td>4/25</td>
<td>Introduction to Spring Framework</td>
<td>In Progress</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>5/2</td>
<td>Project Presentation (Phase III due)</td>
<td>In Progress</td>
<td>Additional Ref.</td>
</tr>
<tr>
<td>15</td>
<td>5/9</td>
<td>Final Exam (open book)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>