CIS 8310: Systems Development

PREREQUISITES:

CIS 3215 or CIS 3270, CIS 8030 or CIS 8130. CSP: I, II, III, IV; V, VI, VII, VIII

Additionally, it is assumed that the students have some familiarity working with computers. Specifically, it is assumed that students be able to, or will learn outside of class, the following:

1. Principles of Programming
2. An object oriented programming language or Integrated Development Environment
3. Creation of web documents
4. Syntax of Java and J2EE or any Microsoft .NET’s Object Oriented web application development languages (C#, VB.NET)
5. Use of MS Office or similar tool to write business reports, academic papers and specification documents
6. Email and a WWW browser
7. Set up a website and upload/download files to/from the site
8. Work effectively in groups

REQUIRED TEXT:


Textbook: Robinson, W.N., Elofson, G., An Introduction to Analysis and Design of Enterprise Software Systems with UML (online draft provided by the instructor/author). This introductory draft bridges theoretical design, UML, and its practical application using common frameworks, such as .NET, J2EE, Hibernate, and content management systems (CMS).


**Textbook (optional for .NET sites):** Developing Application Frameworks in .NET, by Xin Chen, Apress 2004. This book describes the design of an application framework, as well as how to design its reusable components.

**CATALOG DESCRIPTION:**

This course introduces object-oriented software development using an object-oriented programming language such as C++. Emphasis is placed on both object-oriented design and efficient implementation of the design. Topics include: principles of software engineering, management issues, and the prototyping, development, testing, debugging, and maintenance of software systems. The central theme is to build quality software through reuse.

**DETAILED COURSE DESCRIPTION:**

Object-oriented technology is becoming more important because it allows for efficient and effective development of quality software. This course builds on CIS 813 and will provide an in-depth treatment of object-oriented development in C++. Students will learn how to design and develop complex software systems via class projects. Topics include object-oriented design, implementation in C++, developing reusable software components, object-oriented software metrics, and management issues. Modern tools for the design of software systems, graphical user interfaces (GUI), client/server applications, and database design are introduced as implementation vehicles.

**COURSE OBJECTIVE:**

Upon successfully completing this course, a student will:

1. Read and refine an Analysis Document for design.
2. Create and document a logical design.
3. Create and document a physical design.
4. Create and document an application database.
5. Use a CASE tool for systems design.
6. Be familiar with modern object-oriented systems including CASE tools and libraries
7. Use an Application Framework in the design of an information system.
8. Create commonly expected deliverables of systems design: Design Documentation, UML System Design (including UML database design) & System prototypes

**RESEARCH PROJECTS:**

Ideally, when anyone google’s for the following topics, your web site should appear first. (If its in the top 200, then you’ll receive 10 points extra credit. If your site appears in the top 500, then you’ll receive 5 points extra credit.) Start with the Topic Resources, and www.google.com, to find resources that can be linked to your site. (You need not create any content yourself; however, you’re welcome to do so.)

**General Requirements:**
Each presentation shall provide a 15 minute overview of the topic to the class (for presentation dates, refer to the Schedule).
A web site for the topic shall be created. The web site is due (will be evaluated) at the end of the term. (Refer to the Schedule.)

Topics:

2. Information Systems Certification Resources: o IBM, Sun, Microsoft, Cisco, etc. certifications. Student discounts. Does certification make a different?
5. J2EE vs .NET, A Comparison of Application Server Frameworks: Correspondence and difference between major components. How to develop (mainly Microsoft .NET)
6. Microsoft Visual Studio or Eclipse Tips & Tricks: Tips & tricks. What tools (AddIns/PlugIns) are available? Best practices (e.g., organizing projects, solutions, source control)
8. Application frameworks: When does one extend an (open, proprietary) framework vs. when does one build from scratch. Provide links to existing frameworks.
9. DotNetNuke (or other CMS) Development Resources: How to download, install, and develop? What components are available (free, purchase)? Example sites.
10. Outsourcing, How Information Systems Development Compares with other Professions: What is being outsourced? How much? What are the relative rankings? How do disciplines (MIS, Accounting, etc.) compare?

Topic Resources:

1. scholar.google.com
2. www.citeseer.com
3. GSU libraries
5. InformationWeek
6. BusinessWeek
7. Datamation
8. MIS Quarterly
9. Lots of IS journals
10. Intern. Journal of Electronic Commerce
11. Journal of Electronic Commerce
12. Quarterly Journal of Electronic Commerce
13. Internet Computing
14. DATA BASE for Advances in Information Systems
15. Communications of the ACM
16. CIO Magazine
17. Computer World
18. Supply-Chain Council
20. Microsoft case studies
21. InformationWeek
22. IBM SCM

ACADEMIC HONESTY:

Students are encouraged to discuss the assignments out of class and share ideas. However, unless specified as a team assignment, each student must individually complete and submit their own work. The submission of work to be graded that is not your own (which includes any form of "copying") will be considered a violation of the rules of academic honesty.

Abstracted from GSU’s Student Handbook Student Code of Conduct “Policy on Academic Honesty and Procedures for Resolving Matters of Academic Honesty”

As members of the academic community, students are expected to recognize and uphold standards of intellectual and academic integrity. The University assumes as a basic and minimum standard of conduct in academic matters that students be honest and that they submit for credit only the products of their own efforts. Both the ideals of scholarship and the need for fairness require that all dishonest work be rejected as a basis for academic credit. They also require that students refrain from any and all forms of dishonorable or unethical conduct related to their academic work.

Students are expected to discuss with faculty the expectations regarding course assignments and standards of conduct. Here are some examples and definitions that clarify the standards by which academic honesty and academically honorable conduct are judged at GSU.

Plagiarism: Plagiarism is presenting another person’s work as one’s own. Plagiarism includes any paraphrasing or summarizing of the works of another person without acknowledgment, including the submitting of another student’s work as one’s own. Plagiarism frequently involves a failure to acknowledge in the text, notes, or footnotes the quotation of the paragraphs, sentences, or even a few phrases written or spoken by someone else. The submission of research or completed papers or projects by someone else is plagiarism, as is the unacknowledged use of research sources gathered by someone else when that use is specifically forbidden by the faculty member. Failure to indicate the extent and nature of one’s reliance on other sources is also a form of plagiarism. Failure to indicate the extent and nature of one’s reliance on other sources is also a form of plagiarism. Any work, in whole or part, taken from the internet or other computer based resource without properly referencing the source (for example, the URL) is considered plagiarism. A complete reference is required in order that all parties may locate and view the
original source. Finally, there may be forms of plagiarism that are unique to an individual
discipline or course, examples of which should be provided in advance by the faculty member.
The student is responsible for understanding the legitimate use of sources, the appropriate ways
of acknowledging academic, scholarly or creative indebtedness, and the consequences of
violating this responsibility.

*Cheating on Examinations* : Cheating on examinations involves giving or receiving unauthorized
help before, during, or after an examination. Examples of unauthorized help include the use of
notes, texts, or “crib sheets” during an examination (unless specifically approved by the faculty
member), or sharing information with another student during an examination (unless specifically
approved by the faculty member). Other examples include intentionally allowing another student
to view one’s own examination and collaboration before or after an examination if such
collaboration is specifically forbidden by the faculty member.

*Unauthorized Collaboration* : Submission for academic credit of a work product, or a part
thereof, represented as its being one’s own effort, which has been developed in substantial
collaboration with assistance from another person or source, or computer honesty. It is also a
violation of academic honesty knowingly to provide such assistance. Collaborative work
specifically authorized by a faculty member is allowed.
<table>
<thead>
<tr>
<th>Week</th>
<th>Lecture</th>
<th>Reading</th>
<th>Deliverable (points)</th>
</tr>
</thead>
</table>
| 1    | Introduction to Enterprise Design  
*Issues in enterprise business systems.*  
Introduction to patterns: application and program | pp. 1-20  
Farely, Picking a Winner: .J2EE vs. .NET  
TBA | Read the class policies (0) |
| 2    | Creational Patterns | ch. 9 - 13 | Quiz: Patterns I (5) |
| 3    | Structural Patterns | ch. 15 - 21 | Quiz: Patterns II (5) |
| 4    | Behavioral Patterns  
30th: Exam 1: 50 pts | : ch. 22 – 23  
Exam 1 Review | Quiz: Patterns III (5)  
Exam 1: 50 pts |
| 5    | Reverse Engineering: DNA  
*Forward/reverse engineering as a part of the design life-cycle.*  
UML to Code  
*Translation of designs to code.* | ch. 1-8  
*TM: Rose: Reverse-Engineering* | Quiz: UML and Code (10)  
Install Rose XDE (0)  
Install .NET (0); suggestion: install .NET pro & SQL developer |
| 6    | Defining Data [review]  
*Data patterns in UML and ERD.*  
Mapping OO to ERD/Tables, UML 2 DB  
*Mapping classes to a relational database as part of a persistence strategy.*  
Forward/Reverse Engineering: Databases  
*Database generation.* | ch. 2  
Ambler, Mapping Objects to Relational Databases  
pp. 537 - 543 | Quiz: UML and Data (10)  
Exam 2 Review  
Exam 2: 50 pts |
| 7    | Exam 2: 50 pts | | |
| 8    | UML Web Application Extensions  
*A notation to describe client/server pages, client/server scripts --Jim Conallen*  
Data Layer  
*Basic application development* | Modeling Web Applications with UML  
*Rose On-Line Help: Rose Web Modeler  
*UML Extensions for Web Applications: FAQ*  
*Modeling Web Application Architectures with UML* | - |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th>ch. 3-4</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>Transactions1  Transactions2  How to ensure data integrity during operations.</td>
<td>ch. 5</td>
<td>Quiz: Transactions, etc. (10)</td>
</tr>
<tr>
<td>10</td>
<td>Security, .NET Security  Web services, security</td>
<td>ch. 6</td>
<td>Group WWW site (0)</td>
</tr>
<tr>
<td>11</td>
<td>ASP.NET  Group project definition: Project Q/A</td>
<td>ch. 7</td>
<td>Small application (20)</td>
</tr>
<tr>
<td>12</td>
<td>Business (service) Tier</td>
<td>ch. 9</td>
<td>Quiz: Tiers, etc. (10)</td>
</tr>
<tr>
<td>13</td>
<td>Draft presentations (discussion &amp; debugging)</td>
<td>-</td>
<td>Draft Project (25)</td>
</tr>
<tr>
<td>15</td>
<td>Class Presentations</td>
<td></td>
<td>Final Project (50)</td>
</tr>
<tr>
<td></td>
<td>Thursday: Take home <strong>final due</strong> (50)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>