CSC 651: Capstone Project (II)
Syllabus

Dr. Chenglie Hu  Phone: 262-524-7170  e-mail: chu@carrollu.edu
(face-to-face meetings can take place between student and the instructor if desirable)

This is a continuation of the software project that the student has partially finished in CSC650. The student will complete the latter two phases of the Unified Process: Construction and Transition, or an equivalency if another process is used.

This class is the second of two-course sequence that serves as the Capstone for the Masters of Software Engineering. In this course students will:

- Construct or continue evolve the software project he or she has started in CSC650
- Deliver the software product that has been fully tested (and accepted by the client if appropriate)
- Perform continued execution of the Unified Process or another process
- Perform more iteration on the documents produced in CSC650, as well as re-factoring the existing design and/or code
- Produce user manuals and maintenance notes as appropriate

This course, together with CSC650, is meant to evaluate whether the student has mastered the material to meet the goals of the Masters of Software Engineering Program. The Program Outcomes are as follows:

**Carroll College MSE Program Outcomes**
1. Apply sound software engineering principles and methodologies in any software development process regardless of roles they may play as software developers, development leads, or software project managers.
2. Problem-solve (mostly for business problems) at a higher level using enterprise resources, major Web software development frameworks, and sound software design methodologies.
3. Be competitive in making sound judgment on any IT issues that are related to software development.
4. Meet challenges of a software development process as information technologies advance.
5. Be self-motivated and highly effective players in any team environment.

The following objectives for CSC651 will be used to measure student success.

**CSC651 Course Objectives**
Upon completion of the course, the student will

- Demonstrate the ability to understand a business problem and create a solution with appropriate technology (MSE Outcomes 1 - 4)
- Demonstrate the ability of executing the Unified Software/Agile Development Process (MSE Outcomes 1 - 5)
- Demonstrate understanding of object-orientation (MSE Outcomes 1 - 4)
- Demonstrate the proficiency of code writing at professional level (MSE Outcomes 1- 4 )
- Demonstrate the ability of carrying quality software verification (MSE Outcomes 1-4)
- Demonstrate the ability to learn the latest development technologies and frameworks (MSE Outcomes 1-4)

GRADING

This class will be graded by the CSC Faculty and based upon the following criteria.

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>The quality of the completed product, documents produced in the latter two UP phases, the tests execution, and re-factoring [MSE Outcomes 1-5]</td>
<td>90%</td>
</tr>
<tr>
<td>The quality of status report, final presentations and regular communication with the instructor [MSE Goals 1-5]</td>
<td>10%</td>
</tr>
</tbody>
</table>

DELIVERABLES (what will be included in the project binder?)

1. Brief introduction of what has been done in CSC650.
2. All documents that were produced in the Construction and Transition phases.
3. Implementation Issues and Solutions (the issues may include, but not limited to, refactoring the design that you had in CSC650, technical challenges, GUI issues such as user experience and look-and-feel, accommodating additional requirements, use of design patterns or frameworks, issues of using components, etc.)
4. Description of testing performed (unit, regression, and integration testing), the testing results (who did it, how it was done, and what was found...), as well as whether all software requirements have been met.
5. A sample source code that would demonstrate typical business logic you implemented, the code organization, and/or the sophistication of the code.
6. Reflection of the CSC651 experience (quality of the process you've experienced, lessons learned, future plans, maintenance notes, etc).
7. Weekly Journal
9. Technical notes for future maintenance or further development

Tentative Schedule

<table>
<thead>
<tr>
<th>Milestone</th>
<th>Approx. Date</th>
<th>Presentation</th>
<th>Audience</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ground rules and discussion of the course</td>
<td>Week one</td>
<td>Discussion</td>
<td>Lead Faculty &amp; Projects Students</td>
</tr>
<tr>
<td>Status Report on the Construction phase</td>
<td>middle of semester</td>
<td>Informal</td>
<td>Lead Faculty &amp; Projects Students</td>
</tr>
<tr>
<td>Status Report on the Transition</td>
<td>towards the</td>
<td>Informal</td>
<td>CSC Faculty &amp;</td>
</tr>
<tr>
<td>phase</td>
<td>end of semester</td>
<td>Projects Students</td>
<td></td>
</tr>
<tr>
<td>----------------------------</td>
<td>-----------------</td>
<td>-------------------</td>
<td></td>
</tr>
<tr>
<td>Final Presentation</td>
<td>End of Semester</td>
<td>Formal</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Open to All</td>
<td></td>
</tr>
</tbody>
</table>

Note:

- If making code available is not an issue, please burn the code into a CD and place it in your binder.
- For university library archiving purposes, you are expected to compile all relevant documents of CSC651 (except the code) and the some relevant documents form CSC650 (such as requirement specs, SPMP, analysis and design artifacts) into a single PDF file. The submission process will be given later in the semester.

**Final Presentation**

The final presentation is formal. The focus of the presentation should be on the latter two phases of the UP: Construction and Transition, as well as on the demonstration of the finished product.

---

**Policy Statements**

1. The instructor and the College reserve the right to modify, amend, or change the syllabus (course requirements, grading policy, etc.) as the curriculum and/or program require(s).
2. Contact Walter Young Center for disability accommodations
3. ACADEMIC HONESTY: If you significantly use some existing documents or code (not created by yourself), make sure you make an appropriate acknowledgement. For more information refer to the "Academic Dishonesty" policy in VII of the College Student Handbook.