CSC 308 – Software Engineering I

1. CSC 308 - Software Engineering I

2. **credit units** 4  **contact hours** 6

3. **Course Coordinator**: Davide Falessi

4. **Textbook**: Slides and other online material provided by instructor

5. a. **Course Description**: Principles for engineering requirements analysis and design of large complex software systems. Software process models. Methods of project planning, tracking, documentation, communication, and quality assurance. Analysis of engineering tradeoffs. Group laboratory project. Technical oral and written presentations. 3 lectures, 1 laboratory.

   b. **Prerequisite**: CSC 141 or CSC 348.

   c. **Required/Elective/Selective Elective for CPE, CSC, EE, SE**

<table>
<thead>
<tr>
<th>Required</th>
<th>CSC</th>
<th>CPE</th>
<th>SE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Elective</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Selective Elective</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

6. a. **Course Goals/Outcomes**

   The student will be able to:
   
   - Explain the basic concepts of software engineering and the software development process
   - Employ methodological techniques for each of the following software process activities:
     - Requirements: functional requirements, use cases, user stories, non-functional requirements, requirements elicitation from customer, requirements prioritization, requirements estimation, handling requirements changes.
     - Design: Model specification and design: UML Class diagram, Use-case diagram, Sequence diagram. User Interface prototyping. Graphical User Interface (e.g., JavaFX). Decision-making and trade-offs. Design patterns.
     - Coding: Use of a static analyzers (e.g., SonarCloud)
     - Continuous integration (e.g., Travis or Jenkins)
     - Issue tracking (e.g., JIRA or GitHub).
     - Version control (e.g., Git or SVN).
   - Use these methodological techniques in an environment where students obtain ample feedback
   - Develop a specification for a moderate-sized, realistic software system
   - Practice oral and written technical communication skills
   - Practice the art of working effectively in a technical project team
• Use state-of-the-art tools for computer-aided software engineering

b. **How Student Outcomes addressed**
   (“B” = Basic level, “I” = Intermediate level, “A” = Advanced level)

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSC</td>
<td>I</td>
<td>A</td>
<td>I</td>
<td>I</td>
<td>B</td>
<td>A</td>
<td>N/A</td>
</tr>
<tr>
<td>SE/CPE</td>
<td>A</td>
<td>A</td>
<td>I</td>
<td>I</td>
<td>B</td>
<td>B</td>
<td>B</td>
</tr>
</tbody>
</table>

7. **Major Topics Covered: (number of lecture hours each)**
   - Requirements (14)
   - Design (10)
   - Coding (2)
   - Continuous integration (2)
   - Issue Tracking (1)
   - Version Control (1)