



**GRADING:** A = 85-100, B = 75-84, C = 65-74, F = 00-64 (no curving or class-averaging).

**TEXT BOOK:** • Class-notes and other reading material assigned from time to time.

**ACADEMIC INTEGRITY, etc.:**

High standards of academic integrity are expected; plagiarism/cheating on assignments/tests is not tolerated and will be reported to higher authority. Electronic devices (cell-phones, beepers, pagers) are to be turned off. The classroom use of computers is limited to course-related work only to avoid distraction to fellow students and interference with normal classroom activities.

**SERVICE-LEARNING PROJECTS:**

The projects will involve for the most part developing web-based information systems, including databases in some cases. The project requirements will be developed in consultation with the community-partner(s). You will be required to successfully complete and deliver the project to the satisfaction of the community-partner (and the instructor), who will also evaluate your project and assign marks. Expect 4 to 5 site visits (for consultations and demonstrations) to the community partner for the intermediate stages of the projects. Each visit may last upto 30 minutes to 1 hour, depending on the needs.

**REFLECTION COMPONENT:**

- Each student shall write a short "reflection essay" once every 3-weeks based on their experiences, including how they relate to the course objectives. (The instructor will provide guidelines for these essays based on the ORID model.)
- There will be in-class discussion based on these essays to allow students to share their experiences with other students (and the community partner, when possible).

**COURSE OBJECTIVES:**

- Learn how to formulate software projects in consultation with real customers to meet real-life specific needs.
- Learn how to create detailed and verifiable software requirements using domain analysis and how to document them in concise and precise manner using various (static and dynamic) modeling techniques.
- Learn how to analyse software design and how to develop test-plans for a software based on requirements and design; master different testing strategies.
- Learn software process models and role of software management.
- Demonstrate professional behavior in all interactions with the community partners.
- Learn working as a team and master communication skills (written and oral).

**LEARNING OUTCOMES:**

- Understand problems in software design and analysis.
- Understand techniques to solve these problems.
- Understand the gap between the technical aspects of software design and what it means to develop a software that meets a specific customer need, including the role of customer-interaction throughout the software life-cycle (from developing software requirements all the way to the successful delivery of the software).

**CxC CERTIFIED COMMUNICATION-INTENSIVE (C-I) COURSE:**

This is a certified Communication-Intensive (C-I) course which meets all of the requirements set forth by LSU's Communication across the Curriculum program, including

1. Instruction and assignments emphasizing informal and formal [mode 1] and [mode 2];
2. Teaching of discipline-specific communication techniques;
3. Use of draft-feedback-revision process for learning;
4. Practice of ethical and professional work standards;
5. 40% of the course grade rooted in communication-based work; and
6. A student/faculty ratio no greater than 35:1