CSC 4330 - Software Engineering (a service-learning course), Fall-2009

17:10-20:00 M, 232 Coates Hall, LSU

INSTRUCTOR: Dr. S. Kundu 287 Coates, 8-2246 Off. Hrs: 15:30-17:00 MW, or by appointment
kundu@csc.lsu.edu

COURSE DESCRIPTION:

This is a capstone software design project course in Software Design & Development. It will cover major techniques for life-cycle modeling, capturing and modeling software requirements, finite-state and other behavior modeling, and testing large scale software. Also included are techniques for web-designing and relevant modeling techniques.

SERVICE LEARNING:

Service-learning is defined as "a course-based, credit-bearing educational experience in which students (a) participate in an organized service activity that meets identified community needs and (b) reflect on the service activity in such a way as to gain further understanding of the course content, a broader appreciation of the subject discipline, and an enhanced sense of civic responsibility" (Bringle & Hatcher, 1995).

COMMUNITY PARTNERS:

United Methodist HOPE Ministries
Janet C. Simmons, Director of Operations and Social Enterprise
4643 Winbourne Avenue, BR-70808
225-355-0702 (Ext. 22), jsimmons@hopebr.org

Urban Restoration Enhancement Corp
Joyce D. James, Operations Manager
6315 Greenwell St #1, BR-70812
225-356-8871 (Ext. 204), JJames@urecbr.com

Big Buddy Program
Mr. Chancelier D. Skidmore, WordPlay Program Manager
1415 Main St, BR-70802
225-388-9737, bootstrang@aol.com

EXAMS: No make-up exams, except for emergency/sickness (proof required).

45%	Three quizzes (tentative: 21 Sep, 19 Oct, and 16 Nov), each one 15%
40%	Project documentations and other reports (submission by parts, dates to be determined later)
15%	Final project presentation (Nov. 23 and 30). Each student should be prepared to present any part of the project; I shall decide who presents which part.

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

TEXT BOOK: None; use class-notes and other reading material assigned from time to time.

CLASS ATTENDANCE, etc.:

You will loose 5% of the total grade for each missed class unless you provide valid documentation (such as medical reason) and prior notification when possible. Late arrivals by more than 10 minutes is counted as a missed class, and likewise for leaving the class more than 10 minutes before the end of lecture.

Huigh standards of academic integrity are exprected. Plagiarism and cheating on any assignment/tests will not be tolerated.

Electronic devices (cell-phones, beepers, pagers) are to be turned off. The classroom use of computers is limited to course-related work only. Each violation of any of these will cost you 5% of the total grade.

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 bi-monthly "reflection essay" 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.

COURSE OBJECTIVES:

- Develop 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 and how to present them in concise and precise manner using various modeling techniques.
- Learn how to analyse software design and how to develop test-plans for a software based on requirements and design.
- Demonstrate professional behavior in all interactions with the community partners.
- Learn and practice working in groups with peers and how to communicate effectively with both experts and non-experts.
- Develop a sense of how to apply academic learning to solve real-life problems that benefit community needs.

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).

PROJECT DOCUMENTATIONS (and Interim Presentations):

Project title + Abstract	5%
Input-output requirement spec	5%
First Presentation	5%
Test-plan and test-data	5%
Second presentation	5%
High quality coding/programming	5%
Final project document	5%
Final project presentation material	5%
Total	40%

- Total Project Score will be weighted based on your peer evaluation: If the project score is 35 out of 40 and your peer evaluation value is 80%, then your project score is 35×0.80 = 28 out of 40.