Department of Computer Science
College of Engineering,
University of Nevada, Reno

Syllabus CS 426 Senior Projects – 2006

Lectures: TR, 9:30 – 10:45 pm, OSN-202

Instructor: Sergiu Dascalu
Room SEM-236
Tel: (775) 784-4613
E-mail: dascalus@cse.unr.edu
Web: www.cse.unr.edu/~dascalus

Office hours: TR 1:30 – 2:30 pm, SEM-236

Course outline: A continuation of CS425 Software Engineering, this capstone course emphasizes team collaboration and application of modern engineering approaches to software construction. The development by each team of an original, industry-strength software product is the main objective of the course. The instructors will present lectures on the Unified Modeling Language (UML) and its application to object-oriented analysis and design and the teams will report on their project's progress by giving presentations and submitting deliverables related to the project. The teams will deliver and present project parts at the following stages: topic proposal (*concept*), software specification (*requirements*), design (*model*), and implemented software (*final product*). At the beginning of the semester the teams will set up web sites for their projects, web sites that will be then updated regularly to reflect the projects' progress. At the end of the semester there will be a public Senior Projects Workshop with project presentations, demos & posters.

Prerequisite: CS425 Software Engineering

Texts:

- The *required textbook* for CS426 is Jim Arlow's and Ila Neustadt's "UML and the Unified Process: Practical Object-Oriented Analysis and Design," Addison Wesley, 2002, ISBN: 0201770601.
- A *recommended book* for deciding on various human-computer interface aspects of the projects (but not required for the test) is Donald Norman's "The Design of Everyday Things," Basic Books, 2002, ISBN: 0465067107.
- For each individual project *an additional book* ("*project domain book*") will be consulted, together with *at least four reference articles* (journal papers, conference papers, or scientific web publications). This extra reading will be assigned shortly after the project topics will be defined by the teams. The project domain book and the articles will provide support for broadly focused, multi-disciplinary and cross-cultural projects and will be used as references in presentations and project deliverables.

Course website: www.cs.unr.edu/~dascalus/sp2006.html

Initial www pointers:

- The Object Management Group: www.omg.com
- IBM / Rational Software: www.rational.com

Several other addresses of websites that contain project-related resources will be indicated later by the instructors.

Grading scheme (subject to minor modifications):

- Project deliverables: **60%**
  - Concept (**P-I**) [4%]
  - Specification (**P-II**) [10%]
  - Design (**P-III**) [11%]
  - Implementation, integration, and testing (**P-IV**) [35%]
- Project presentations and publications **18%**
  - Presentations (specs, design, workshop) (**PRES-I, II, III**) [8%]
  - Project website (**PWEB**) [6%]
  - Poster (**POST**) [4%]
- Midterm examination (**TEST**) **15%**
- Class participation (scheduled classes & workshop, **WS**) **7%**

Grading scale:

A	90 -100	[maximum 100]
A-	87 - 89	
B+	84 - 86	
B	79 - 83	
B-	76 - 78	
C+	73 - 75	
C	68 - 72	
C-	65 - 67	
D+	61 - 64	
D	56 - 60	
D-	50 - 55	
F	< 50	

Notes on grading:

- Requirements for grade A: at least 90% overall, at least 85% in class participation, and at least 60% in test
- There are no make-up tests or homework in this course

Late submissions:

Late submissions of assigned work will be penalized with a deduction of 10% of the grade per late day, to a maximum of two late days for each submission. No material will be accepted after two days past the deadline. For example, a project deliverable that is worth 90/100 points will receive 90*0.9 = 81/100 points if it is one day late. The same deliverable will receive 90*0.8 = 72/100 points if it is two late days and it will not be accepted if it is more than two days late. Note that late days are not divisible in subunits. Late days are not allowed for presentations, poster, and test.

On plagiarism and cheating:

Plagiarism and cheating will not be tolerated. It will be dealt with according to the policies of the University of Nevada, Reno regarding academic dishonesty. Please read these policies at www.unr.edu/stsv/acdispol.html

Legal notices on the world-wide web:

When accessing www resources such as downloadable software, technical reports, papers, on-line tutorials, etc., do not forget to read their accompanying legal notices and comply with their provisions.

Tentative Schedule CS 426 Senior Projects Spring 2006

Week	Period	Contents
1	Jan 23 – Jan 27	Course goals, outline and organization, project topics, work plan.
2	Jan 30 – Feb 03	Lectures on UML and UP: Requirements Workflow, Analysis. Definition of project topics. **Project teams set (Feb 02)**
3	Feb 06 – Feb 10	Project group meetings **Project concept deliverable due (P-I, Feb 10)**
4	Feb 13 – Feb 17	Lectures on UML and UP: Design. Draw for presentations order. **Project website set up (PWEB, Feb 15)**
5	Feb 20 – Feb 24	Lectures on UML & UP: Design Guidelines for peer evaluation. **Project specification deliverable due (P-II, Feb 22)**
6	Feb 27 – Mar 03	**Project specification presentations (PRES-I, Feb 28 & Mar 02)**
7	Mar 06 - Mar 10	***Invited lecture*** & Lecture on UML and UP: Implementation **Project design deliverable due (P-III, Mar 08)**
8	Mar 13 – Mar 17	Project group meetings
9	Mar 20 – Mar 24	**Spring break, no classes**
10	Mar 27 – Mar 31	Lecture on UML and UP: Implementation Recap for midterm exam
11	Apr 03 – Apr 07	***Invited lecture*** & **Midterm exam (TEST, Apr 06)**
12	Apr 10 – Apr 14	**Project design presentations (PRES-II, Apr 11 and Apr 13)**
13	Apr 17 – Apr 21	**Pre-demos (Apr 18 and Apr 20)** (Project group meetings)
14	Apr 24 – Apr 28	***Invited lecture*** & Workshop preparation **Posters due (POST, Apr 26)**
15	May 01 – May 05	**Project implementation, integration, and testing deliverable & Internal project demos (P-IV, May 01)**
	May 05	**Workshop presentations, demos, and posters (WS, PRES-III)**

Accreditation outcomes:

Preliminary notes:

* ABET criteria for engineering programs (a-k) are listed after outcomes on page 6.
** Computer science program objectives (1-4) are listed after ABET criteria on page 7.
*** Course components are abbreviated as in sections Grading Scheme on page 2 and Tentative Schedule on page 4. In addition, talks by invited professionals from industry are denoted IT.

AO_1: The students will collaborate on a team project to deliver an industry-strength application and will extend their knowledge of requirements elicitation and specification, software design, implementation, and integration. Also, they will increase their ability to work towards accomplishing project goals as team members.

Strategies and Actions used to produce the outcome:

- Form a group project
- Propose a project idea (project concept)
- Build a project website
- Develop the project following software engineering phases: requirements elicitation and specification, design, implementation, integration, and testing
- Invited talks by experienced professionals from the software engineering industry

ABET Criteria* covered: a, c, d, e, g, k
Program Objectives** covered: 2, 3, 4
Main course components*** for this outcome: P-I, P-II, P-III, P-IV, P-WEB, Invited talks

AO_2: The students will enhance their skills of communicating project requirements and designs with software engineers and other project stakeholders in a standardized manner, using the UML (Unified Modeling Language) for software development.

Strategies and Actions used to produce the outcome:

- Class lectures on advanced UML concepts
- Midterm test focused on UML as applied to practical project development
- Preparation of project deliverables that emphasize communicating specifications and designs using UML

ABET Criteria covered: a, c, d, e, g, k
Program Objectives covered: 2, 3, 4
Main course components for this outcome: Lectures, TEST, P-II, P-III

AO_3: The students will know how to communicate project specifications, designs, and implementations in both writing and speaking. They will also increase their experience in reviewing others' work.

Strategies and Actions used to produce the outcome:

- In-class presentations by students of project concepts and deliverables (specifications and designs)
- Internal project demo presented by students to the instructors

- Anonymous peer reviews of project parts presented in class by the project teams

ABET Criteria covered: d, g
Program Objectives covered: 4
Main course components for this outcome: Pres-I, Pres-II, P-Web, P-I, P-II, P-III, P-IV, Peer reviews, Class participation

AO_4: The students will formally present the completed project work to an audience of varied technological background.

Strategies and Actions used to produce the outcome:

- Preparation and delivery by each team of a formal public presentation of the team's project (Workshop presentation, at the end of the semester)
- Preparation of a 4 x 3 feet poster that describes the results of the team's project work (also to be presented at the Senior Projects Workshop)
- Setting up a project website, updated regularly during the semester
- Peer review and feedback from the audience at the public Senior Projects Workshop

ABET Criteria covered: d, g, i
Program Objectives covered: 2, 4
Main course components for this outcome: Pres-III, P-WEB, P-POST, WS, Peer reviews, WS audience feedback

ABET criteria:

a. an ability to apply knowledge of mathematics, science, and engineering
b. an ability to design and conduct experiments, as well as to analyze and interpret data
c. an ability to design a system, component, or process to meet desired needs
d. an ability to function on multi-disciplinary teams
e. an ability to identify, formulate, and solve engineering problems
f. an understanding of professional and ethical responsibility
g. an ability to communicate effectively
h. the broad education necessary to understand the impact of engineering solutions in a global and societal context
i. a recognition of the need for, and an ability to engage in life-long learning
j. a knowledge of contemporary issues
k. an ability to use the techniques, skills, and modern engineering tools necessary for engineering practice

Computer science program criteria:

Our graduate students will have achieved […]:

1. a broad general education assuring an adequate foundation in science and mathematics relevant to computing.
2. a solid understanding of concepts fundamental to the discipline of computer science.
3. good analytic, design, and implementation skills required to formulate and solve computing problems.
4. the ability to function and communicate effectively as ethically and social responsible computer science professionals.