CMSI 699
INDEPENDENT STUDIES:
SOFTWARE ENGINEERING PRACTICUM
http://myweb.lmu.edu/dondi/spring2008/cmsi699

Spring 2008 — Doolan 106
Variable sessions and meetings, 3 semester hours
Office Hours: TR 3–6pm or by appointment

John David N. Dionisio, PhD
e-mail: *dondi@lmu.edu,* AIM: *dondi2LMU*
Doolan 106; (310) 338-5782

Course Objectives

To embark on a self-directed course of study in a specific area of computer science, under the guidance of the instructor. The student selects the topic and assists in identifying source material. In addition to learning more about the specific topic, the student will learn how to organize, execute, and document an intensive, individualized semester of self-study. Additional objectives depend on the selected subject matter.

Course Requirements

Since the target of this software engineering practicum is the implementation of OpenGL "pass-through" functionality for Web browsers that use the WebKit open source project, the student must have mastery of the C, C++, and JavaScript programming languages, proficiency with computer graphics algorithms and how they are presented and implemented in OpenGL, and familiarity with Web technologies and standards. Some knowledge of the Objective-C programming language may also be helpful.

Materials and Texts

To be identified and reported by the student.

Course Work and Grading

Graded coursework consists of 1 study journal in the form of a blog (20%), 1 final study report (40%), and committed contributions to the WebKit open source project in support of the desired OpenGL pass-through functionality (40%). Letter grades are determined as follows: ≥ 90% gets an A– or better; ≥ 80% gets a B– or better; ≥ 70% gets a C– or better. The instructor may curve your grade upward based on qualitative considerations such as degree of difficulty, effort, time constraints, and overall attitude throughout the course. Grades are never curved downward.

Study Journal

You will maintain a Web log (blog) of your study progress. No particular blog service, server, or software is required; you may select whatever is most available or convenient at the time. Your blog serves as an on-going account of your study. Blogging gives you an opportunity to express yourself while ideas or events are still fresh, at the same time providing structure and persistence that gives it lasting value.

Your study journal will be graded according to the same criteria as the final paper (see below). The frequency and number of blog entries affects the content and organization components of the study journal's grade.

The study journal will be graded at the end of finals week, May 9. You are, of course, free to maintain the blog beyond the semester — the spirit of a 699 course is, after all, to give you an opportunity to get official credit for something that you would want to study on your own anyway. Continuing the blog provides you with a framework for pursuing this study beyond the scope of the semester.

Final Study Report

You will formally document the overall result of your studies in the form of a final report to be submitted at the end of the semester. The report shall consist of at least the following sections:

1. An introduction that states the background and motivation for this course of study,

2. A literature review describing the source materials studied, and

3. A technical description of the software that you have written: an overview of its design and context (i.e., existing code into which it was integrated, if applicable), followed by any relevant implementation details and possible future work, with a focus on what was learned during the semester.

To help you to focus on the actual work and content of the report (as opposed to busy work such as formatting and reference management), it must be written using LaTeX. We will talk about LaTeX early in the semester. A LaTeX outline/template for the report will be provided to get you started.

There are no hard limits on length, but 10–20 pages in LaTeX's default article format, not including the list of references cited, is typical. The report will be evaluated along the following criteria:

1. *Content (40%):* What is the quality of the work? Are the background and motivation relevant and well-stated? Is the literature review thorough and well-described? Is the summary or survey complete and substantive? How well-documented is the programming project?

2. *Organization (30%):* Is the text structured well? Are its ideas and flow easy to follow? Are distinct sections or topics clearly identified?

3. *Writing (20%):* Are statements clear and easy to follow? Is the language precise and grammatically correct? Is the paper's tone appropriate?

4. *Polish (10%):* Is the content properly proofread? Are there any misspellings, typos, or other formatting faux pas?

The final study report is due at the end of finals week, May 9.

Open Source Contributions

You will apply what you learn in the form of committed contributions to the WebKit open source project. These contributions should be clearly identified for review and grading.

The project will be graded along these criteria:

1. *Design (30%):* How good is the overall structure of the code? Is it clear, flexible, and easy to maintain? Is it elegant or innovative? How well does it apply the principles of "separation of concerns" and "one change, one place?"

2. *Functionality (30%):* How well does the code work? Does it fulfill requirements? Are its results accurate or correct? Does it perform its tasks in a reasonable amount of time? How well do unit tests validate the code?

3. *Naming (20%):* Are program entities — classes, subroutines, variables, etc. — clearly and consistently named? Do their names correspond to their functions and roles?

4. *Comments (15%):* Are comments provided where appropriate? Are they clear and well-written? Does the code take advantage of any special support for comments provided by the project language or platform (e.g., JavaDoc)?

5. *Version control (5%):* Is the code committed at reasonable intervals? Are milestones appropriately tagged? Are adequate descriptions of provided in the commit logs?

WebKit contributions as of the end of finals week, May 9, will serve as the basis for this grade.

Attendance

Meeting and session schedules are determined individually, and may vary according to the specific subject matter and/or course work. Remember that the university add/drop with 100% refund deadline is January 18. The deadline for withdrawal or credit/no-credit status is March 14.

University Policy on Academic Honesty

Loyola Marymount University expects high standards of honesty and integrity from all members of its community. Applied to the arena of academic performance, these standards preclude all acts of cheating on assignments or examinations, plagiarism, forgery of signatures or falsification of data, unauthorized access to University computer accounts or files, and removal, mutilation, or deliberate concealment of materials belonging to the University Library.

Course Schedule

Dependent on the specific subject matter and ongoing progress.