CSCI 334: System Design and Implementation

Syllabus – Spring 2010

Ben Coleman
coleman@cs.moravian.edu
214 PPHAC

Office Hours: M-F 10:10 - 11:10
or by appointment
Office Phone: 610-625-7781

Course Description

A project-oriented study of the ideas and techniques required to design and implement a computer-based system. Topics include project organization, design, documentation, and verification.

Course Goals

Upon completion of this course, a successful student will be able to:

- Design large software projects using patterns and other standard techniques.
- Communicate both orally and in writing with clients, colleagues, and supervisors.
- Document specifications and code using standard tools.
- Work in a team to design and develop software projects.

Required Texts

In addition to the following required texts, supplementary readings will be given periodically during the semester.

- *Object-Oriented Software Engineering Using UML, Patterns, and Java*, third edition by Bernd Bruegge and Allen H. Dutoit
- *The Pragmatic Programmer* by Andrew Hunt and David Thomas

You should expect to spend about an hour before each class session working through the readings. This means reading the text for detail and working to learn vocabulary – not just skimming through the material before class.

Graded Material

Below is a brief description of each of the assignments for the course. In class I will hand out detailed descriptions of the requirements and grading guidelines.

- **Letter of Application to the Course** – You will write a formal letter requesting admission into the course. The purpose of this assignment is to begin our discussion of targeted technical writing. In class we will review drafts of this letter, and then you will produce a final submission.

- **Cover Letter and Resume** – In this assignment you will create the appropriate materials to submit for a job application. You will select the job to which you apply. You will submit multiple drafts of these documents, but only the final version will be graded.

- **Journal Assignments** – Informal journals will be used to explore ideas from the Pragmatic Programmer text - one for each of the chapters of the book that we cover.

- **Project Performance** – Throughout the semester we will be working on jAmaseis, seismologic software with a target audience of seismology educators. Our clients for this application are members of the Incorporate Research Institutions for Seismology (IRIS) Education and Outreach (E&O) group. In addition to the three planned assignments (below), your overall performance on the project will be assessed by the other students in the course and by me. This grade is not based on whether or not the project is completed successfully, but on your level of contribution. You will not be involved in every aspect of the project, but you are expected to contribute in a timely manner when given assignments. As evidence of your involvement, you will maintain a work log throughout the semester.

- **Project Email** – Early in the project we will communicate through email with the IRIS E&O group. In this assignment, you will each draft and edit your own version of the email. This will be turned in and graded, and then as a class we will combine the versions into a single email to send to IRIS.

- **Web Documentation** – Throughout the semester we will incorporate additional documentation to the jAmaseis web page. This assignment will be completed by the group through multiple drafts. Only the final product will be graded, and all students will receive the same grade.

- **Final Analysis Paper** – Instead of a final exam, you will write a final paper due to me by 11:30 on Friday, May 7. In this paper, you will critique the project and discuss how the content of the course was utilized in the project.

- **Tests** – Two tests will be given during the semester on Friday, March 5 and Friday, April 23. You may only re-schedule a test for college-approved absences or documented illness. In either case, you must contact me *before* the beginning of the test.

Grade Determination

- Letter of Application to the Course – 5%
- Cover Letter and Resume – 10%
- Journal Assignments – 15%
- Project Performance – 20%
- Project Email – 5%
- Web Documentation – 10%
- Final Analysis Paper – 15%
- Tests – 20%

All grades will be computed on the standard scale using plusses and minuses

Course Policies

- **Responsibilities** – Your attendance is expected at each class meeting. You are also responsible for the contents of reading assignments, handouts, class activities, and class email.
- **Late Policy** – I understand that life sometimes gets in the way of getting work done. Consequently, late assignments will be accepted without penalty in the class after the assignment was due. However, this policy should not be used as a crutch, and if you frequently use it I will deduct from your grade. After the next class session, late work will not be accepted unless there are exceptional circumstances.
- **Extensions** – In a similar vein, I am generous with extensions on work if you approach me *before* the day the assignment is due.
- **Academic Honesty** – Except on tests, you are *encouraged* to discuss the material and work with other students in the course. This policy does not allow you to copy another student's work verbatim – you must produce your own code or write-up of the material. Work together to learn the concepts, but keep in mind that you are ultimately responsible for the material on the tests.
- **Disabilities** – If you have a disability that may affect your performance in this course, please contact me immediately to discuss academic accommodations.

Course Schedule

NOTE: OOSE refers to the text *Object-Oriented Software Engineering*, and PP refers to *The Pragmatic Programmer*.

Date	Reading	Topic
M Jan 18		• Day 1 Activities
W Jan 20	• OOSE Ch 1: pp. 3-25	• Introduction to Software Engineering
F Jan 22	• OOSE 2.1-2.4.2: pp. 29-59	• UML Use Cases and Class Diagrams
M Jan 25	• OOSE 2.4.3-2.5: pp.59-72	• UML Interaction and Activity Diagrams
W Jan 27	• OOSE 3.1-33: pp. 77-92	• Project Organization
F Jan 29	• OOSE 3.4-3.6: pp. 92-114	• Project Communication
M Feb 1	• PP Preface: pp. *xvii-xxiv* • PP Chapter 1: pp. 1-23	• The Pragmatic Philosophy
W Feb 3	• Handout	• Personality
F Feb 5	• OOSE 4.1-4.4: pp. 121-148	• Requirement Elicitation
M Feb 8	• OOSE 4.5: pp. 148-153	• Managing Requirements Elicitation
W Feb 10	• OOSE 4.6: pp. 153-168	• Case Study
F Feb 12	• PP Ch 2: pp. 25-69	• A Pragmatic Approach
M Feb 15	• OOSE 5.1-5.4: pp. 173-199	• Use Case to Objects
W Feb 17	• OOSE 5.5-5.6: pp. 199-218	• Managing Analysis
F Feb 19	• PP Ch 7: pp. 201-222	• Before the Project
M Feb 22	• OOSE Ch 6: pp. 223-254	• System Design: Decomposing
W Feb 24	• OOSE 7.1-7.4: pp. 259-284	• System Design: Addressing Goals
F Feb 26	• OOSE 7.5: pp. 284-289	• Managing System Design
M Mar 1	• OOSE 7.6: pp. 290-302	• Case Study
W Mar 3	• PP Ch 4: pp. 107-136	• Pragmatic Paranoia
F Mar 5		• Test #1
M Mar 7 – F Mar 12		• Spring Break
M Mar 15	• OOSE 8.1-8.4: pp. 307-337	• Object Design: Reuse
W Mar 17	• OOSE 8.5-8.7: pp. 337-345	• Managing Reuse
F Mar 19	• PP Ch 5: pp. 137-170	• Bend or Break
M Mar 22	• OOSE 9.1-9.4: pp. 349-375	• Object Design: Interfaces
W Mar 24	• OOSE 9.5-9.6 : pp. 375-388	• Managing Design
F Mar 26	• OOSE 10.1-10.4: pp. 393-422	• Mapping Models to Code

Date	**Reading**	**Topic**
M Mar 29	• OOSE 10.5-10.6 : pp. 422-432	• Managing Implementation
W Mar 31	• PP Ch 6: pp. 171-199	• While You are Coding
F Apr 2	• OOSE 11.1-11.4: pp. 437-476	• Testing
M Apr 5		• Easter Break
W Apr 7	• OOSE 11.5: pp. 476-486	• Managing Testing
F Apr 9	• OOSE 12.1-12.4: pp. 493-526	• Rationale Management
M Apr 12	• OOSE 12.5: pp. 526-532	• Managing Rationale
W Apr 14	• OOSE 13.1-13.4: pp. 537-566	• Configuration Management
F Apr 16	• OOSE 13.5: pp. 566-571	• Managing Configuration Management
M Apr 19	• OOSE 14.1-14.4: pp. 575-611	• Classis Project Management
W Apr 21	• OOSE 14.5: pp. 611-616	• Agile Project Management
F Apr 23		• Test #2
M Apr 26	• PP Ch 8: pp. 223-259	• Pragmatic Projects
W Apr 28	• OOSE Ch 15: pp. 621-647	• Software Life Cycle
F Apr 30	• OOSE 16.1-16.4: pp. 651-680	• Methodologies

The details of this syllabus and schedule are subject to change based on our progress through the material.