

Wright State University

CORE Scholar

Computer Science & Engineering Syllabi

College of Engineering & Computer Science

Spring 2013

CS 7140: Advanced Software Computer Engineering

Prabhaker Mateti

Wright State University - Main Campus, prabhaker.mateti@wright.edu

Follow this and additional works at: https://corescholar.libraries.wright.edu/cecs_syllabi



Part of the [Computer Engineering Commons](#), and the [Computer Sciences Commons](#)

Repository Citation

Mateti, P. (2013). CS 7140: Advanced Software Computer Engineering. .
https://corescholar.libraries.wright.edu/cecs_syllabi/692

This Syllabus is brought to you for free and open access by the College of Engineering & Computer Science at CORE Scholar. It has been accepted for inclusion in Computer Science & Engineering Syllabi by an authorized administrator of CORE Scholar. For more information, please contact library-corescholar@wright.edu.



CS 7140: Advanced Software Engineering

[Prabhaker Mateti](#)

1. CS 7140 - Advanced Software Engineering (Catalog Description)

[3 Credit Hours] This course covers advanced topics in software engineering. Aspects of problem specification, design, verification, and evaluation are discussed. We will focus on design methods, including software patterns and software architecture, plus some advanced topics involving formal methods of software specification or evaluation using software metrics. Students will participate in team projects to apply the methods discussed. Prerequisites: CEG 4110 or CEG 6110.

CEG 4110 - Intro to Software Engineering

Concepts of software engineering. Analysis, design, and implementation of software engineering concepts that comprise structured programming and design. Case studies serve as examples illustrating the software life-cycle model. Prerequisites: CS 3100

CS 3100 - Data Structures and Algorithms

Study of the implementation of data structures and control structures in professional computer programs. Introduction to the fundamentals of complexity and analysis. Study of common standard problems and solutions (e.g., transitive closure and critical path). Emphasis on high-level language software design. Prerequisites: CS 1181 and MTH 2530 and MTH 2570

2. Source Materials

1. Course Home Page: <http://www.cs.wright.edu/~pmateti/Courses/7140/> Bookmark this link and visit it periodically. All course related announcements will be posted on this page.
2. Text Book: There is no prescribed text book. URLs to Required Readings will be given during the course.
3. Reading List for this course is a transitive closure starting from the course home page. Items marked as Require Reading are sources for exam questions. Other items do enhance your understanding, but, you may skip.

3. Course Content

The following is tentative. The order of topics will be roughly as indicated, except for project related discussions. The numbers in parentheses at the end of each topic is an estimate of the number of (75-minute) lectures I intend to devote. I lecture on only the conceptually difficult sections, leaving the rest as reading assignments.

1. Programming in the Small, Large, Giga; Metrics (2)
2. Assertions, Proofs v Testing (2)
3. Project on Maintenance; Tools: eclipse, idea, git (2)
4. Functional Specifications, Performance Specifications (2)
5. Design: Abstractions, Object Oriented Design, Very High Level Languages (2)
6. Formal Methods: assert.h, splint, frama-c (2)
7. Famous Examples of Spec/Design (2)
8. Testing: Coverage, Unit, Integr (2)ated; Famous Examples of Bug Analyses (2)
9. Refactoring and Design Patterns (2)
10. Secure Programming (2)
11. Software Systems Architecture (2)
12. Program Understanding (2)
13. Trends: agile-, eXtreme- Programming (2)

4. Grading

Class and Newsgroup Participation 5%

Please participate in the class room discussions. Feel free to discuss openly, to ask what may appear to be dumb questions, and to catch any errors that I make. Be alert, study the backup material right away, and solve the exercises promptly.

There is a local discussion group (see the link on the course home page) that you should read and participate regularly as a vehicle for discussing your solutions. I will read this group regularly, and note silently who is contributing what. I expect other students to be able to find flaws and fix them. Only when the most fundamental errors are being made will I break my silence and contribute a response.

Exams 20+25%

There will be two exams contributing 20% plus 25% to the final grade. The date for the mid-semester exam will be announced in class. The date for the final exam is as set by the Registrar. Both exams are closed-book, and comprehensive.

Project 30%

Project work in this course is done by small teams (of perhaps 4 students). You will be given the source code files of a working program written in either C++ or Java. You will be asked to critique it and make various improvements. There are 6 tech reports (Requirements, Specifications, Design, Implementation, Testing, and Final Report), 5% each, due at various stages of the project. The due dates are on the course home page. A description of the project will be announced later.

Homework 20%

There are two homework assignments, each 10%. These are to be done individually. The homework will be given piece meal and it is expected that you will work on them as and when given without waiting for a due date. The eventual due date will be announced on the course home page.

There is a penalty of 5% per day, week end counted, for late submissions.