Software Engineering

Computer Science 420 - Fall 2013 - Ms. Katz
Tuesday and Thursday 10:00 AM - 11:50 AM in Roddy 256

Instructor
- Ms. Beth Katz - katz@cs.millersville.edu - http://cs.millersville.edu/~katz
- Office: Roddy 139
- Office Hours: 9 - 10 AM and noon to 1 PM Tuesday and Thursday and by appointment
  email is an effective way to contact me - katz@cs.millersville.edu or bethkatz@comcast.net
- Phone: 871-2269 (office on class days) or 872-5740 (home; 8 AM to 8 PM only)
- Class Web Page: http://cs.millersville.edu/~katz/cs420.html

Prerequisites
You are expected to have earned a C- or better in both CS 330 and CS 362 before taking this course.

Textbook
There is no required textbook for this course. There will be numerous required readings.

Course Description and Goals
Software Engineering is the process of developing high-quality software using sound engineering techniques. Our main goals this term are to recognize the qualities of software, to study the major activities involved in creating software, and to practice using some of the tools and techniques used in developing high-quality software.

The course provides an overview of Software Engineering organized around the major activities of software development - planning and project management, specification, design, implementation, testing, and evolution. Topics include the goals of each activity, a survey of techniques that are used, and hands-on experience with some of those techniques.

In addition, this course will provide the student with experience in teamwork by incorporating a major project developed by a group of students using techniques learned in the course. This project will involve significant writing, programming, and presentation/speaking activities as the students specify, design, implement, and test their solutions. Team members will present their solutions to the class throughout the term which will provide students an opportunity to prepare and deliver a technical presentation.

Grading
Midterm (October 10) = 20%
Term-long team project (many documents) = 40%
Individual homework, presentation concerning project, 2-page ethics analysis paper = 15%
Final exam (December 11, 2:45pm - 4:45pmm) = 25%

I use a ten-point grading scale (93 = A; 90 = A-; 87 = B+; 83 = B; ...). You must attempt all examinations and participate cooperatively in the team project to pass the course. You must complete the individual assignments by yourself. The team project is the only collaboration allowed.

Attendance
I expect you to attend class regularly, keep up with the reading, and participate in class discussions. Excessive absences tend to result in lower course grades. You should expect to meet with your team frequently outside the scheduled class time.

Academic Honesty
You may ask me questions at any time. Email is particularly good for this. Copying or collaboration with anyone else on the individual assignments or tests is not permitted and may result in failure in the course. You are expected to collaborate extensively within your team on the team project, but the individual assignments should be completed on your own. Refer to the university policy on academic honesty for details and penalties. I report academic dishonesty to the provost.
Tests
Tests will be in the classroom. They will be cumulative with closed book, notes, and neighbor. The only allowed tools are pen or pencil, blank paper, and your brain. If you miss a test, you will receive a grade of zero for that test. Exceptions may be made at my discretion. Contact me immediately, preferably before the exam, if you have a problem.

Individual Assignments
There are no late assignments. You should complete them by yourself. However, you may ask me questions.

Ethics Analysis Paper
After we discuss the ACM / IEEE Computer Society Software Engineering Code of Ethics and Professional Practice, you will write a 2-page analysis of a situation. Your grade on this paper will be based on both your ethical analysis and your writing. This paper must be your own work. It will be due November 26th.

Presentation
Each student will give a 10-15 minute presentation concerning some aspect of the team project. This presentation will not be a demo. It may not include execution of your project. It should be a well-organized technical presentation. You must hand me an outline of your talk before you start. Each student will be graded individually on the presentation although they may have help in preparing the details and the documents used in the presentation.

Team Project
A significant part of the hands-on activity in this course involves a term-long team project. You will be assigned to a 4-5 person team. You are expected to participate fully in this project and perform tasks allocated to you by the team. The team will decide on due dates within some constraints and deliver that schedule as part of its initial project plan. The team must meet that schedule but may request extensions for good reasons.

While building a high-quality piece of software is important, the process of building it is just as important. You must use the tools and techniques we will discuss.

You must collaborate within your team, but some parts will be done individually or in pairs. In particular, coding should not be an all-team-eyes-on-the-code activity.

Communicating clearly within your team and in the documents you create will help you succeed. The documents will help you communicate within your team at a higher level than code. You will revise these documents as you learn new techniques and are further into your project. Spend the time and effort to develop and use them well. In the long run, that effort will pay off.

Altogether, the project counts as 40% of the grade. The project components will count roughly the amount shown below. I may change this but hope not to. Each part will have multiple sub-parts. For example, there will be an initial project plan and more detailed project plans later in the term. Each iteration will have multiple document deliveries. Grading in different iterations will emphasize different phases. For example, implementation will be more important in iteration 1, specification and design will be more important in iteration 2, and specification, verification, and testing will be more important in iteration 3.

Project plans = 3%
Iteration 1 = 11%
Iteration 2 = 14%
Iteration 3 = 12%

The project is not so big that you will spend a lot of time writing code. It is deliberately small enough that you can concentrate on the process of development. In many ways, the project is too small to need some of the tools we will use. But that helps you apply them to a project you understand.

Gaining experience in building high-quality software is our goal.