Programming Languages and Software Engineering

Computer Science 330 - Spring 2014 - Ms. Katz
Tuesdays and Thursdays 8 AM - 9:50 AM in Roddy 147

Instructor

- Ms. Beth Katz - katz@cs.millersville.edu - http://cs.millersville.edu/~katz
- Office: Roddy 139
- Office Hours: 10 AM to noon 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)
- Class Web Page: http://cs.millersville.edu/~katz/cs330.html

Prerequisites

You are expected to have earned a C- or better in both CS 140 and CS 162 before taking this course.

Required Textbook

You can borrow or share. It won't be a reference during tests. You don't need it in class.

Course Description and Goals

This course investigates the relationship between the development of software engineering and the development of modern programming languages. The structure and vocabulary of modern programming languages are studied. Programming language topics include: binding, data control and sharing, type checking, subprograms and parameters, and implementation of language constructs. Major software engineering paradigms are studied. Software engineering topics include: requirements definition, specification, design, verification, validation, and relationships of paradigms to languages.

Grading

Test 1 (February 27) = 25%
Test 2 (April 8) = 25%
Final exam (May 6, 2:45-4:45pm) = 25%
Assignments (about six including a 5-page paper and presentation) = 25%
Papers will be due April 22nd.
Short presentations will be April 29 and May 1.

I use a ten-point grading scale (93 = A; 90 = A-; 87 = B+; 83 = B; ...). You must attempt all examinations and assignments to pass the course. You must complete the programming and laboratory assignments by yourself unless explicitly told otherwise. Pair programming, as allowed on specific assignments as noted, is the only collaboration allowed. I report academic dishonesty to the provost.

Attendance

I expect you to attend both lecture and lab, read the text as noted, and participate in class discussions. I will take roll. Getting notes from a friend is not sufficient. Tests emphasize what we have covered in class. If you miss more than three classes without an excuse, I may lower your course grade by a letter grade.

Academic Honesty

You may ask me questions at any time. Email is particularly good for this. Copying or collaboration with anyone else on assignments or tests is not permitted and may result in failure in the course. Refer to the university policy on academic honesty for details and penalties. I report academic dishonesty to the provost.

You may discuss approaches to solving the problem as well as general aspects of the technique we are studying. However, if you can't explain why you did something, you probably didn't do it yourself. Finding or purchasing a solution on the internet or elsewhere and turning that in as your own work is also plagiarism. If you do have reason to use code from the book or another source, include attribution in a comment in your program. When in doubt, ask me.

We will use pair programming on some, but not all, assignments. Details when needed.
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.

Assignments and Labs
I'll usually provide a description of the lab activity the class period before the lab. I expect you to read through the handout before you come to lab. This will make the lab period less stressful and a more effective use of our time. You are expected to remain in the lab (or classroom) for the full class period or until you have successfully completed all work. Most assignments will require further work outside the lab period.

There are no late programs. A program must be delivered by the time and date specified for the assignment to my katz330 account. If your program is not complete, submit what you have for partial credit. Some points are better than zero points. Develop your programs incrementally, and you will always have something to turn in for partial credit.

All programs must pass through the g++ compiler on the MU CS department's Linux computers without compiler errors with the -Wall option. Programs that do not compile have a maximum grade of 0.

A program that compiles cleanly is not necessarily correct. Part of this course is software engineering, and I expect you to use appropriate software engineering techniques. I will deduct points for missing, incomplete, and incorrect features. Be sure your program meets the problem specification. Read your programs for errors and test them with several suitable test sets. I am extremely good at breaking programs. Break your program before I do.

I also read your programs. I will deduct points for poorly indented and commented programs as well as those that use poor programming practices or do not use the techniques and data structures specified in the assignment. ‘A’ does not mean adequate. Refer to example programs from class.

The paper will be about 5 pages, and I expect the writing to be college-level technical writing. There will be more details on this when assigned. The presentation will be short (a few slides and several minutes).