

CS601 Principles of Software Development

Terence Parr

Fall 2014

Class Administration

- * Prof. Terence Parr
- * Office: HR528
Email: parrt@cs.usfca.edu
<http://parrt.cs.usfca.edu>
- * Hours: anytime my door is open, by appointment, or by email
- * TA: Bonkers the cat



About me



- * I am a programmer, not a theoretician
- * My PhD is in computer engineering not computer science, but I have focused on computer language design and application for 30 years
- * Designed and built ANTLR, StringTemplate
- * Cofounded and created jGuru (1996-2004)
- * Broad experience as a consultant and employee



Goal

- ✳ Make you a better programmer
- ✳ Prepare you for graduate school
- ✳ Make you an employable commercial programmer

Course Format

- * Class periods of 1:05min each 3 times per week for 15 weeks
- * Instructor-student interaction during lecture is encouraged.
- * All programming will be done in the Java programming language.
- * Midterm, Final (2nd exam)
- * Multiple small projects and one large: gmail

Grading

- * Projects 55%
- * Midterm exam 20%
- * Final exam 20%
- * Quizzes, class participation, mini-assignments 5%

Late policy

- * There is no such thing as a late project. That's a 0.
- * Unless you are sick or have a family emergency, I will not change deadlines for projects or exam times.

Grades

- ✳ "A" grade is above and beyond what most students have achieved
- ✳ "B" grade is an average grade for a graduate student or what you could call "competence" in a business setting.
- ✳ "C" grade means that you either did not or could not put forth the effort to achieve competence.

Strict grading

- * Projects that do not run exactly as specified will lose 10% of the total points.
- * All products graded on a UNIX machine
- * Do not hardcoded filenames
- * UNIX filenames are case-sensitive as are Java symbols
- * All class and method signatures must be correct
- * Standard input versus program argument

Academic honesty

- * You must abide by the copyright laws of the United States and academic honesty policies of USF
- * You may not copy code from other current or previous students
- * Small snippets of code from the web is usually okay, but in general you may not use code you pull from the web. Please ask before doing so.
- * At the very least, you must provide references for code you use
- * The golden rule: **You must never represent another person's work as your own**
- * First time: 0 on the project or exam. Second time: failure of the course

Disabilities

- * If you are a student with a disability or disabling condition, or if you think you may have a disability, please contact USF Student Disability Services within the first week of class, or immediately upon onset of the disability
- * Reasonable accommodations are made for legitimate disabilities

Rough outline

- * Part I -- Technology, mechanics of programming
 - * OO, I/O, threads, debugging, protocols, sockets, services, web apps, databases.
- * Part II -- Design and Development strategies
 - * Testing, re-factoring, top-down design, agile, extreme, patterns, case studies

Books

- * There is no formal book for the class, but you will be asked to read articles and book excerpts.
- * One of the most common ref'd books will be: Code Complete (2nd edition) by Steve McConnell.
- * We will discuss Frederick Brooks' Mythical Man Month as well.
- * Maybe The Clean Coder by Robert Martin.

Software

- * I recommend IntelliJ for Java and WebStorm for JavaScript/TypeScript
- * Submission of projects will be via git / github
- * MySQL, mongodb, SQLLite, or similar
- * Web stuff: Jetty webserver, AngularJS, Bootstrap

My expectations

- * Lots of coding!
- * You must learn how to learn. Reading code and APIs and articles is part of your job
- * You must learn problem solving
- * Try to solve it first before you ask me
 - * My first question: what did you find on Google?
 - * But, don't waste 3 days trying to solve something I can solve in three minutes
- * I'm happy to explain how to solve your problem or discuss software development in general; it's my job. Come see me

The debugging squirrel

- * “My program doesn’t work” is meaningless; be precise
- * Explain your problem to the squirrel
- * If you can get past the debugging squirrel, I’ll debug your code

