Introductions

- Instructor, Prof. James Bieman
  - Phone: (970) 491-7096.
  - Office: Room 462, CS Building
  - Office Hours: TBA.
  - Email: Bieman@CS.ColoState.EDU
- Teaching Assistant, Fatmaya Assiri.
  - Email: fatmaya@cs.colostate.edu
  - Phone: 970-491-TBA
  - Lab Hours: TBA
  - cs314@cs.colostate.edu goes to both of us.
- Lectures: Mon/Wed/Fri, 10:00 - 10:50 AM.

Focus: Methods to Develop Commercial-scale Software.

- Large size systems: may be millions of lines long.
- Large development teams.
- Large or highly specialized user communities.

Large-scale software development → Large-scale problems

Static connections in a small system

Detailed Call-Graph of CIV

Shows 3% of the entire call graph.

CS314 Objectives

You will learn to
- use software models (in UML) to
  - analyze software requirements,
  - express designs, and
  - guide implementations;
- apply systematic code testing techniques;
- use well-known software design patterns.
Objectives (continued)

You will learn to use software development tools for:
- program development,
- configuration management (revision control),
- software modeling,
- testing.

Objectives (cont.):

You will learn
- a variety of software development processes, the software lifecycle, and their implications;
- software safety challenges;
- ethical issues in software engineering.

A Software Safety Challenge
The Google Driverless Car

Prerequisites

- CS253 and all the prerequisites for CS253 including:
  - Discrete structures (CS166, or CS160-CS161) and
  - Data structures (CS200).
- You should know how to program using object-oriented programming languages and structures.
- Programming in CS314 will be (mostly) done in Java.

Course Materials

- Required Text: Software Engineering custom text from the CSU bookstore.
- Course Notes.
- Supplemental material as needed.

Software Development Tools

- Eclipse
- Testing tools:
  - Junit 4: partially installed on Eclipse. You will need to add a link and put it on the "build path".
  - Emma and/or EclEmma (Eclipse plugin) test coverage tools.
- Revision control:
  - Git or Github
  - Alternatives include Subversion (svn) or Subclipse (svn Eclipse plugin).
- Modeling tools:
Additional Materials

- Articles from technical journals, magazines, and web sites.
- More Reference Texts:

Learning Activities

Learn vocabulary and practice through:

- Lectures,
- Reading: text and other references,
- Exams,
- Quizzes
- Assignments,
- Online discussions,
- In-class design studios.

Exams

- Expect 3 exams:
  - Midterm Exam 1
  - Midterm Exam 2
  - Final exam.
- Closed book and closed notes.
  Exception: you may bring one 8.5 x 11 inch sheet of paper with your notes to the exam. You must put your name on the sheet and turn it in with your exam.

Quizzes

- Quizzes are based primarily on reading material.
- Complete reading assignments before class.
- Quizzes are usually given at the start of class; they take approximately 10 minutes.
- Expect one (or more) quizzes per week.

Assignments

- Work in pairs (teams of 2) on all assignments.
  Except Assignment 0, which is posted.
- Planned assignments involve unit testing, programming, design, modeling, system testing, design patterns, requirements, and software safety.

Discussions

- Five discussion topics:
  Topic 1
  “What makes programming challenging?”
- Post your views on the topic, or a response to another posting. Professional behavior is required.
- Discussions last for 1 week.
- Up to 10 Discussion Points point per discussion.
Design Studios

• Apply concept from lectures.
• Groups (3 or 4 students) work on specific problems.
• Must hand in solutions to receive points.
  Groups will present solutions for some design studios.
• Pre-lab preparation required for some design studios.

Web Support

• Calendar, Bulletin Board, Discussions.
• Password protection: view grades.
• Most assignments are submitted via Canvas.
• Check the class Canvas bulletin board daily for important announcements.
• Assignments are posted to Canvas
  • Key links:  
    – Weekly Guide
    – Class Session Blog
    – Syllabus

Course Rules

• Keep up.
  – Attend lectures.
  – Be on time.
  – Start assignments early.
• Behave professionally.
  – Put cell phones on silent.
  – Leave classroom to answer if necessary.
  – Be polite on discussion boards and email.
  – Respect others.
• Laptops in back row or rows.

Grading

Approximate breakdown:
• Assignments: 30% of grade.
• Midterm exam 1: 15%.
• Midterm exam 2: 15%
• Final Exam: 20%
• Quizzes: 5% of grade
• Design Studios: 10% of grade
• Discussions: 5% of grade

Policies

• All work done on time
  – 48 hour grace for assignments with a 10% penalty.
  – Exceptions only for unforeseeable circumstances.
• Re-grade Policy
  - The GTA grades assignments, and is the first contact for grading concerns. Contact the instructor if you still disagree.
  - Bring up grading concerns promptly (within 5 days).

Outline of Course Topics

2. Object-Oriented Concepts: connection between designs and implementation.
4. Object-Oriented Architecture & Design
5. Requirements analysis.
7. What we’ve missed.