

• Lecture 1: Course Overview

- Kenneth M. Anderson
 - January 11, 2005
-

• CAETE Announcements

- **In-Class Students**
 - CAETE has a busy studio schedule
 - Be sure to exit promptly so next class can begin on time
 - Food and Drink are not technically allowed
 - Drinks are tolerated, as long as you keep the studio clean!
- **Live-Site Students**
 - Place the speakerphone away from and TV and point it away from the TV
 - If you have connection problems
 - hang up, wait 15 seconds, call again
 - If your speakerphone has a mute button, use it when not talking to reduce background noise
- **Distance Students**
 - Textbooks can be ordered from the CU Bookstore
 - Call 303-492-3444 or 800-255-9168
- **NOTE: due dates for CAETE students are one week behind due dates for in-class students**

• Class Participation

- I expect you to participate!
- Questions
 - "Stupid Questions" — No such thing!
 - "Clarification Questions" — Please do!
- Discussion
 - "Silent Tomb" — Not allowed! Not fun for you or me!
 - Feel free to interrupt me!
 - Feel free to ask me to slow down
 - I tend to talk fast
- CAETE Students
 - Live-Site Students (see above)
 - Distance Students (via e-mail)

• The Instructor

- Ken Anderson
- Office Hours
 - Right after class: 11 AM to 12 PM on Tuesdays and Thursdays
 - or by appointment
- E-mail: kena@cs.colorado.edu
- Phone: 303-492-6003
- Mailing Address
 - Dr. Kenneth M. Anderson
 - University of Colorado
 - Department of Computer Science
 - 430 UCB
 - Boulder CO 80309-0430
- Fax: 303-492-2844

• My Background

- Assistant Professor
 - Fourteenth Semester at CU
 - Fifth time teaching CSCI 6448
 - Research Interests
 - Open Hypermedia and Web Technologies
 - Software Engineering
 - Requirements Traceability
 - Workflow
 - Design of Software Systems
 - Software and Design Experience
 - Systems
 - Metis Workflow System for Digital Libraries, versions 1.0 and 1.1
 - InfiniTe: Information Integration Environment, versions 1.0 and 2.0
 - Themis Structural Computing Environment, versions 1.0 and 2.0
 - Chimera Open Hypermedia System, versions 1.0-4.0
 - C2 Architectural Style
 - Chiron-1 User Interface Development System
-

• Teaching Philosophy

- "sage on stage" vs. "guide at your side"
- Answering Questions
 - Sometimes the answer will be "I don't know!"
- "Hands-On" Assignments
 - Assignments tend to make you apply what you have been learning in lecture
 - In class "assignments": occasionally I will stop lecture and have you try a technique out in class
 - CAETE students please stop the "tape" at that point and do the exercise at home

• Useful URLs

- CS Department
 - <http://www.cs.colorado.edu/>
 - Instructor's Website
 - <http://www.cs.colorado.edu/users/kena/>
 - Class Website
 - <http://www.cs.colorado.edu/users/kena/classes/6448/s05/>
-

• About the Class Website

- You have one continuous homework assignment this semester:
 - Check the class website EVERY day
 - Preferably more than once each day
- Class website will be your source for
 - Class schedule
 - Homework assignments
 - Lectures
 - Pointers to class-related information

• Prerequisites

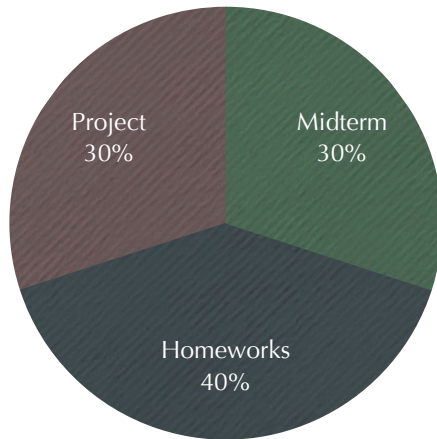
- Background in basic software engineering concepts
 - Software systems and tools
 - Software engineering theory (e.g. The Mythical Man-Month by Fred Brooks)
 - Software Life-Cycles
 - Requirements, Design, Implementation, Testing, Deployment
- Experience with at least one object-oriented programming language
 - I will NOT be teaching object-oriented programming in this class
 - If you need to learn OOP:
 - Head First Java by Bert Bates and Kathy Sierra
 - Thinking in Java by Bruce Eckel

• Course Topics

- Basic/Advanced OO Concepts and Terminology
- OO Analysis and Design Techniques
 - CRC Cards
 - Requirements-Driven Design
 - Domain-Driven Design
- Design Patterns
- Refactoring
- UML

• Assessment

- You will be evaluated based on your work in three categories



- **Homeworks**
 - Designed to make you apply the techniques discussed in class
- **Midterm**
 - Used to evaluate your knowledge of the topics covered in the first half of the semester
- **Project**

- Group or individual project that applies the requirements and design techniques covered in class to create a small object-oriented system

• Notes on Assignments

- **Electronic Submission via Moodle**
- **Text or Postscript/PDF formats only**
 - I don't have every application under the sun!
 - Its up to you to learn how to create Postscript/PDF files (PDF has been around for ten years; Postscript longer)
- **Late Penalty**
 - Late assignments will be assessed a 20% late penalty
 - Late assignments can be submitted up to one week late
 - If you know you are going to be late on an assignment, contact me BEFORE the assignment is due

• Assignment Headings

- Please place the following information on the first page of all assignments
 - Student Name
 - Course Number
 - Company Name (if applicable)
 - Assignment Name
-

• Course Textbooks

- Object Design: Roles, Responsibilities, and Collaborations
 - Rebecca Wirfs-Brock and Alan McKean
- Domain-Driven Design: Tackling Complexity in the Heart of Software
 - Eric Evans
- NOTE: In addition, I will be drawing material from many other books throughout the semester

• Class Goals

- Learn how to model "problems" and "solutions" using object-oriented concepts and techniques
 - The former task is also known as "requirements"
 - The latter task is known as "design"
- Learn practical techniques for improving the quality and structure of your code
 - both before its written (software architecture / design patterns)
 - and after its written (refactoring)
- Gain a working familiarity with the Unified Modeling Language
 - I won't teach UML directly, but I'll be using UML notation in my examples

• Importance of Requirements and Design

- Requirements = Understanding a problem within a problem domain
 - "Problem Domain" where the "domain" in "Domain-Driven Design" comes from
 - A problem domain can have many different problems within it
 - we need to figure out which specific problem(s) we are addressing
- Design = Creating a solution to the problem
- This is an age-old problem: getting to a solution from a set of requirements
 - Fred Brooks classifies the difficulties encountered while working on this problem as "the essential difficulties of software development"
 - NOTE: OO techniques are just one way of doing requirements and design

• Homework 1

- Submit via the Moodle
 - your background, including your technical skills
 - why you are taking this class
 - what you are hoping to learn
- See class website for complete details
- Note: You will need to register with the Moodle before you can submit this assignment