

Object-Oriented Analysis and Design

Kenneth M. Anderson
University of Colorado, Boulder
CSCI 4448/5448 — Lecture 1 — 08/25/2009

A bit about me...

- Associate Professor
- At CU since July 1998
- Ph.D. at UC Irvine (1997)
- Research Interests
 - Software Engineering
 - Hypermedia
 - Web Engineering



A little bit more...

- 23rd Semester at CU
- 8th time teaching CSCI 4448/5448 (used to be called 6448)
- Software Development Experience
 - Approximately 16 systems, 30K-100K LOC each
 - Some industry experience with IBM & Unisys
 - Experience with academic/industry collaboration
 - NCAR, NREL, Northrop Grumman, ioSemantics

Office Hours

- ... by appointment
- When meeting with me, we'll use either
 - ECCS 127 (across from the CSEL)
 - Faculty Lounge (across from ECOT 717)

Class Website

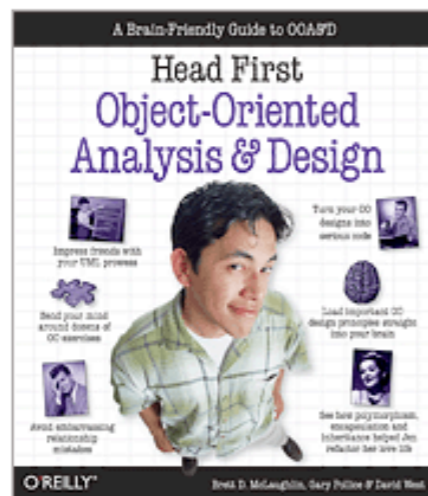
CSCI 4448/5448 – Fall 2009

Object-Oriented Analysis & Design

[Home](#) [What's New](#) [Textbooks](#) [Evaluation](#) [Syllabus Statements](#)

Textbooks

There are two textbooks for the class.



[Head First Object-Oriented Analysis & Design](#) by Brett D. McLaughlin, Gary Pollice & David West. Published by O'Reilly. ISBN-10: 0-596-00867-8. This book offers an excellent introduction to the topic of OO A&D. It introduces the topic gradually making use of code-based examples that illustrate clearly the relationship between decisions made at the analysis and design level (often via the use of UML diagrams) and their impact on the code of an object-oriented system. It also clearly defines analysis techniques in a way that make them feel "more real" to students who may not have been exposed to analysis and design considerations

before this class.

[<http://www.cs.colorado.edu/~kena/classes/5448/f09/>](http://www.cs.colorado.edu/~kena/classes/5448/f09/)

About the Class Website

- Check the website *every day!*
 - An RSS feed of the What's New page is available
- The website is your source for
 - class schedule
 - homework assignments
 - announcements
 - etc.

Textbooks



Head First Object-Oriented Analysis and Design



Head First Design Patterns

Head First Series

- Lots of examples
 - Including tight integration of UML and code
- Covers fundamental concepts well
 - Some may question “less than academic tone” but they touch on all the concepts of “more academic” textbooks (with better examples)
- Employs multiple strategies for teaching concepts
- Repetition, Repetition, Repetition!

Class Participation

- I welcome participation by students
 - Feel free to interrupt me during lecture to ask questions!
 - I'm going to work this semester at making lectures more of a conversation
- Stupid Questions — No such thing!
- No participation leads to “silent tomb” — Boring!
- If I'm speaking too fast, stop me and tell me to slow down!

Teaching Philosophy

- “sage-on-stage” vs. “guide-at-your-side”
- Answering questions
 - Sometimes the answer will be “I don’t know!”
- Current Tech
 - I try to incorporate references to current technology in my teaching
 - Ruby, Python, Ruby on Rails, Django, etc.
 - If you encounter a cool new OO-related technique or language feel free to send a pointer to me and I’ll try to talk about it during lecture

Bias?

- I don't use Microsoft Windows and so have no experience with
 - .Net
 - C#
 - F#
 - etc.
- However, I'm not "anti-Microsoft" and I will welcome student presentations on Microsoft technology

Goals of the Class

- Provide students with knowledge and skills in:
 - object-oriented concepts
 - OO analysis, design, and implementation techniques
 - object-oriented design methods
 - (aka software development life cycles)
- Students should view OO software development as a **software engineering process** that has well-defined stages with each stage requiring specific tools and techniques

Course Structure

- First 8 weeks
 - Cover fundamental OO A&D concepts
 - objects, UML, use cases, analysis & design techniques
- Second 8 weeks
 - Cover additional design and implementation techniques
 - design patterns, refactoring, testing, concurrency, etc.

Course Evaluation


- Midterm (30%)
- OO Framework Analysis and Presentation (30%)
 - Teams of 2 to 3 people
- Class Project (30%)
 - Teams of 3 to 4 people
- Homework assignments (10%)
 - May involve programming, solving a design problem, answering questions about the assigned reading, etc.
 - 5 to 6 assignments spread across the semester

Programming Languages (I)

- Examples will be in Java, Python and Ruby
- OO Programming is NOT a central topic of the class
 - If you have never programmed using an OO language, start working your way through a tutorial:
 - Java: <<http://java.sun.com/docs/books/tutorial/>>
 - Ruby: <<http://tryruby.hobix.com/>>
 - Python: <<http://docs.python.org/tut/>>
 - Many more available online; use your favorite search engine!
- I believe that analysis and design are the HARD parts of OO development

Programming Languages (II)

- Assignments

- You may use any OO language (within reason) when working on your assignments
 - If you pick C#, I may have to meet with you to see you run your code.
 - Please no Object-Oriented Perl! 

Submitting Assignments

- Assignments will be submitted via e-mail and will vary according to format
 - Text: submitted within the body of an e-mail message
 - PDF: submitted as an attachment of an e-mail message
 - .zip or .tar.gz: send a link in e-mail and we'll download the file
- Adopting this approach as ITS will sometimes “nuke” messages with .zip attachments
 - such that recipients are not notified that someone tried to send a message and senders are not notified that their message was not delivered!

Honor Code

- I encourage collaboration in this class via the team-based projects and homeworks
- Only the midterm (obviously) is to be worked on individually
- The Student Honor Code applies to classes in all CU schools and colleges. You can learn about the honor code at:

<http://www.colorado.edu/academics/honorcode/>

Late Policy

- Assignments handed in late incur a 20% penalty
- Assignments can be handed in up to ONE week after the initial due date (except for the final assignment of the class project)
 - after that you are out of luck...

Syllabus Statements

- The University asks that various statements be presented to students at the start of each semester
 - Disability Accommodations
 - Religious Observances
 - Classroom Behavior
 - Discrimination and Harassment
 - Honor Code
- These statements are on the class website at:

<http://www.cs.colorado.edu/~kena/classes/5448/f09/ss.html>

Discussion

- Survey of Class
 - Experience with OOP?
 - Experience with OO A&D?
 - Definitions?
 - Do you have any topics that you'd like to explore?
- Any other questions?

Ken's Corner (Explanation)

- A section of lecture to provide fun pointers to class-related information
 - Some items may be related to software engineering in general
- Will include
 - Quirks and/or cool features of OO programming languages
 - Pointers to topical SE/OO articles
 - Things You Should Know[®]
 - but do not appear in your textbooks

Ken's Corner

- The best way to learn about software engineering, design, and programming is to read the blogs of excellent software developers
- Here are a few that I recommend
 - Joel on Software: <<http://www.joelonsoftware.com/>>
 - Tim Bray's ongoing: <<http://www.tbray.org/ongoing/>>
 - Wil Shipley's Call Me Fishmeal: <<http://www.wilshipley.com/blog/>>
 - One of the best developer-written posts EVER
 - <<http://ridiculousfish.com/blog/archives/2006/05/30/old-age-and-treachery/>>
 - Alas this blog is now (mainly) inactive.
 - Finally, to see developers blow off steam, check out:
 - The Daily WTF: <<http://thedailywtf.com/>>



Coming Up Next

- Lecture 2 and 3: Introduction and Review of Fundamental Object-Oriented Concepts
 - Read Appendix 2 of the OO A&D book
- Lecture 4: Great Software
 - Read Chapter 1 of the OO A&D book