Introduction

CSCI 4448/5448: Object-Oriented Analysis & Design

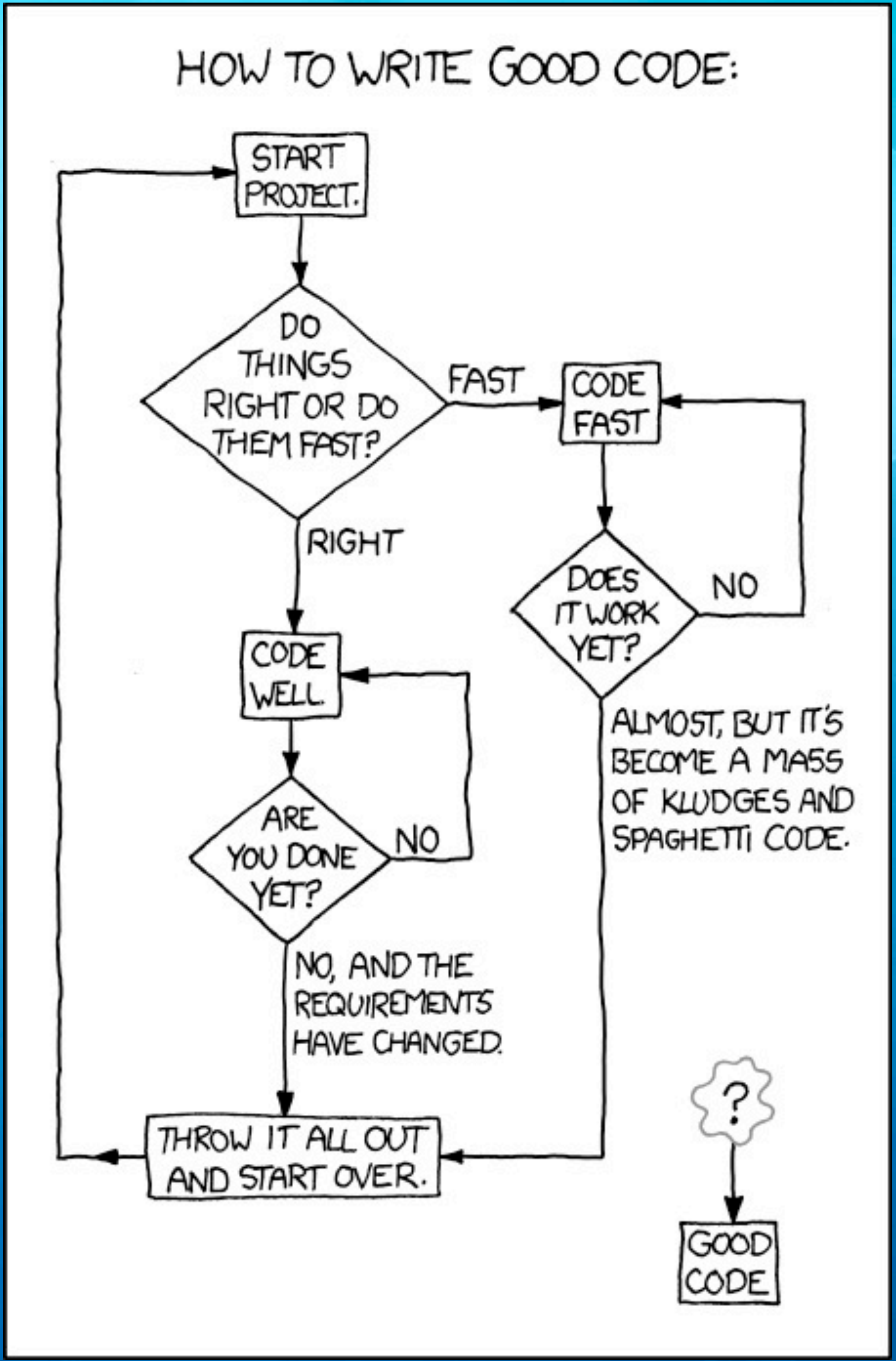
Lecture 1 — 01/11/2011

This class aims to teach you a style of software design that can enable you to reach the box labelled "Good Code" in the diagram on the right.

Software Design is not completely a black art... there are design techniques that lead to better results when applied in support of creative expression.

From the excellent web comic, xkcd:

<http://xkcd.com/844/>


HOW TO WRITE GOOD CODE:
START PROJECT.
DO THINGS RIGHT OR DO THEM FAST?
FAST
CODE FAST
DOES IT WORK YET?
NO
ALMOST, BUT IT'S BECOME A MASS OF KLUDGES AND SPAGHETTI CODE.
RIGHT
CODE WELL.
ARE YOU DONE YET?
NO
NO, AND THE REQUIREMENTS HAVE CHANGED.
THROW IT ALL OUT AND START OVER.
?
GOOD CODE

About Me

- Associate Professor
  - Ph.D. at UC Irvine (1997)
  - 12.5 Years at CU
- Ninth Time Teaching This Class
- Research
  - Software Engineering
  - Hypermedia & The Web

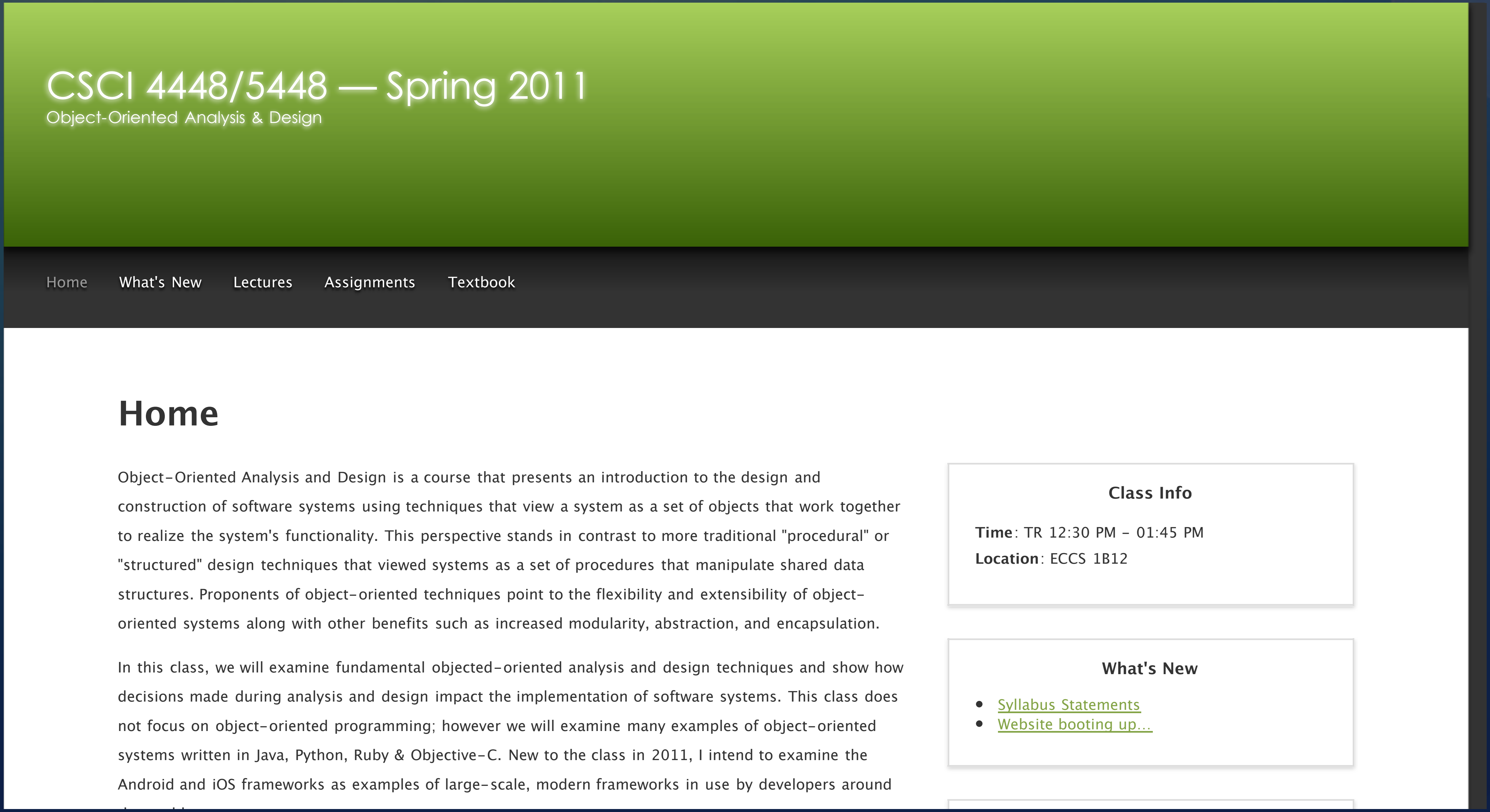
Office Hours

- Thursdays 2 PM to 3 PM
- ECCS 111 (Down the hall from the CSEL)

- Please send me e-mail to let me know you'll be attending

- You can also meet with me at other times by sending e-mail to make an appointment http://www.cs.colorado.edu/~kena/classes/5448/s11/

Class Website



CSCI 4448/5448 — Spring 2011
Object-Oriented Analysis & Design

Home

Object-Oriented Analysis and Design is a course that presents an introduction to the design and construction of software systems using techniques that view a system as a set of objects that work together to realize the system's functionality. This perspective stands in contrast to more traditional "procedural" or "structured" design techniques that viewed systems as a set of procedures that manipulate shared data structures. Proponents of object-oriented techniques point to the flexibility and extensibility of object-oriented systems along with other benefits such as increased modularity, abstraction, and encapsulation.

In this class, we will examine fundamental objected-oriented analysis and design techniques and show how decisions made during analysis and design impact the implementation of software systems. This class does not focus on object-oriented programming; however we will examine many examples of object-oriented systems written in Java, Python, Ruby & Objective-C. New to the class in 2011, I intend to examine the Android and iOS frameworks as examples of large-scale, modern frameworks in use by developers around

Class Info

Time: TR 12:30 PM - 01:45 PM
Location: ECCS 1B12

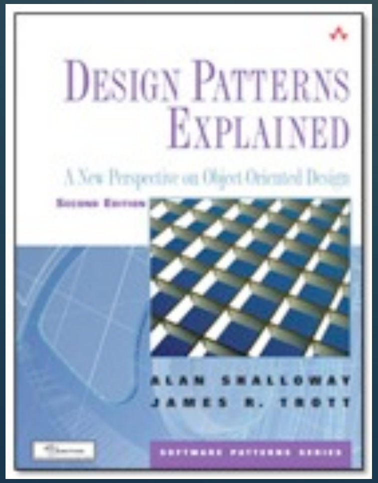
What's New

- Syllabus Statements
- Website booting up...

Check the Website Everyday!

- There is an RSS feed associated with the What's New page to make this easy for you to do!
- The website is your source for
  - the class schedule
  - homework assignments
  - announcements
  - etc.

Textbook


DESIGN PATTERNS EXPLAINED
A New Perspective on Object-Oriented Design
Second Edition
ALAN SHALLOWAY
JAMES R. TROTT
SOFTWARE PATTERNS SERIES

- Design Patterns Explained
  - A New Perspective on Object-Oriented Design, Second Edition
- Alan Shallloway and James R. Trott
- Addison Wesley, © 2005

- Book discusses a design methodology that encourages the use of design patterns early in a software development effort
- I will also be drawing on other resources throughout the semester

Teaching Philosophy

- I want you to participate!
  - Feel free to interrupt me when you have a question
  - Feel free to tell me to slow down if I'm speaking too fast
- I will try to learn your name (although, with more than 90 students, it's going to be tough!!)
- Learning by Doing
  - I will try to create conversations each lecture and will also insert in-class activities where appropriate
  - Homeworks will ask you to apply techniques learned in class

Goals of the Class

- Provide students with knowledge and skills in:
    - object-oriented concepts
    - OO analysis, design and implementation techniques
    - OO design methods (software 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
- Gain some experience with the Android and iOS frameworks

Course Structure (Tentative)

- Weeks 1 - 4: Chapters 1 - 11 of the Textbook
- Weeks 5 - 7: Introduction to Java, Objective-C, Android and iOS
- Week 8: Midterm; Exact Date: Tuesday, March 1, 2011
- Weeks 9 - 10: Intermediate and Advanced Android and iOS
- Spring Break
- Weeks 11 - 13: Chapters 12-25 of the Textbook
- Weeks 14 - 15: Object Relational Mappings (Hibernate); Dependency Injection (Spring); Project Presentations

Course Evaluation

Note: Grading standards will be higher for graduate students.

- Undergraduates
  - Midterm (30%)
  - Homeworks (65%)
  - Class Participation (5%)
- Graduate Students
  - Midterm (30%)
  - Presentation (30%)
  - Homeworks (35%)
  - Class Participation (5%)

Homeworks will include a class project that can be worked on in teams of 2 to 4 people. The "presentation" for graduate students will address an advanced topic of OO A&D or OO Programming or introduce an OO Framework in depth. Presentations will appear on the website and "executive summaries" will be presented in class.

Honor Code

- I encourage collaboration in this class via the homeworks (which will include a semester project); You may work on them in teams of 2 to 4 students
- All students must work on the midterm individually (obviously)
- Graduate students must work on their presentations 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/>

Submitting Assignments

- Assignments will be submitted via e-mail and will vary via 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 a message that contains a .zip attachment
    - When they do this, the message simply disappears!
    - I don't receive it and you don't know it wasn't delivered!

Late Policy

- Assignments submitted after the deadline incur a 15% penalty
  - meaning the maximum grade on a late assignment is a B
- Assignments can be submitted up to one week after the initial due date (except for the final assignment of the class)
  - after that you are out of luck…

Syllabus Statements

- The University asks that various policies be presented to students at the start of each semester. These policies include
  - Disability Accommodations
  - Religious Observances
  - Classroom Behavior
  - Discrimination and Harassment
  - Honor Code
- See <http://www.cs.colorado.edu/~kena/classes/5448/s11/syllabus-statements.html> for more details

Discussion (I)

- How many people have used an object-oriented programming language before?
    - Java? C#? C++? Objective-C? Python? Ruby? Javascript? Others?
- What features make a language object-oriented?

Discussion (II)

- How many people are comfortable starting from scratch and creating:
    - a script?
    - a desktop application?
    - a web service?
    - a mobile application?
    - a system of systems? (i.e. desktop plus web service)
    - a database-backed application?

Discussion (III)

- When you create a program from scratch:
  - do you use OO techniques?
  - OO design heuristics?
  - design patterns?
- If not, what style of software design do you use?
  - What styles of software design are you aware of?

Discussion (IV)

- What is design?
- What comes before design?
- What comes after design?
  - Does this question make sense in software development?
- What would make the process of software design object-oriented?

Programming Languages

- Examples will be written in Java, Objective-C, Python and Ruby
- OO Programming is NOT a central topic of the class
  - This stance stems from my view that analysis and design are the hard parts of OO software development
  - However, I will be devoting lectures to introduce Java and Objective-C
- Assignments
  - Note: You will be required to write some homework assignments in the Java language, otherwise any OO language may be used

Bias?

- I do not have much experience with C++, C# or .Net
    - As a result, I do not include examples of these two languages or the .Net framework in my lectures
- However, I am not "anti-Microsoft" or "anti-C++" and therefore welcome student presentations on C++ or Microsoft technologies

Coming Up Next

- Lecture 2: The OO Paradigm
  - Read Chapter 1 of the Textbook
- Homework 1: To be assigned on Friday
- Lecture 3: UML
  - Read Chapter 2 of the Textbook
- Lecture 4: Example problem and traditional OO solution
  - Read Chapters 3 and 4 of the Textbook