Course Overview

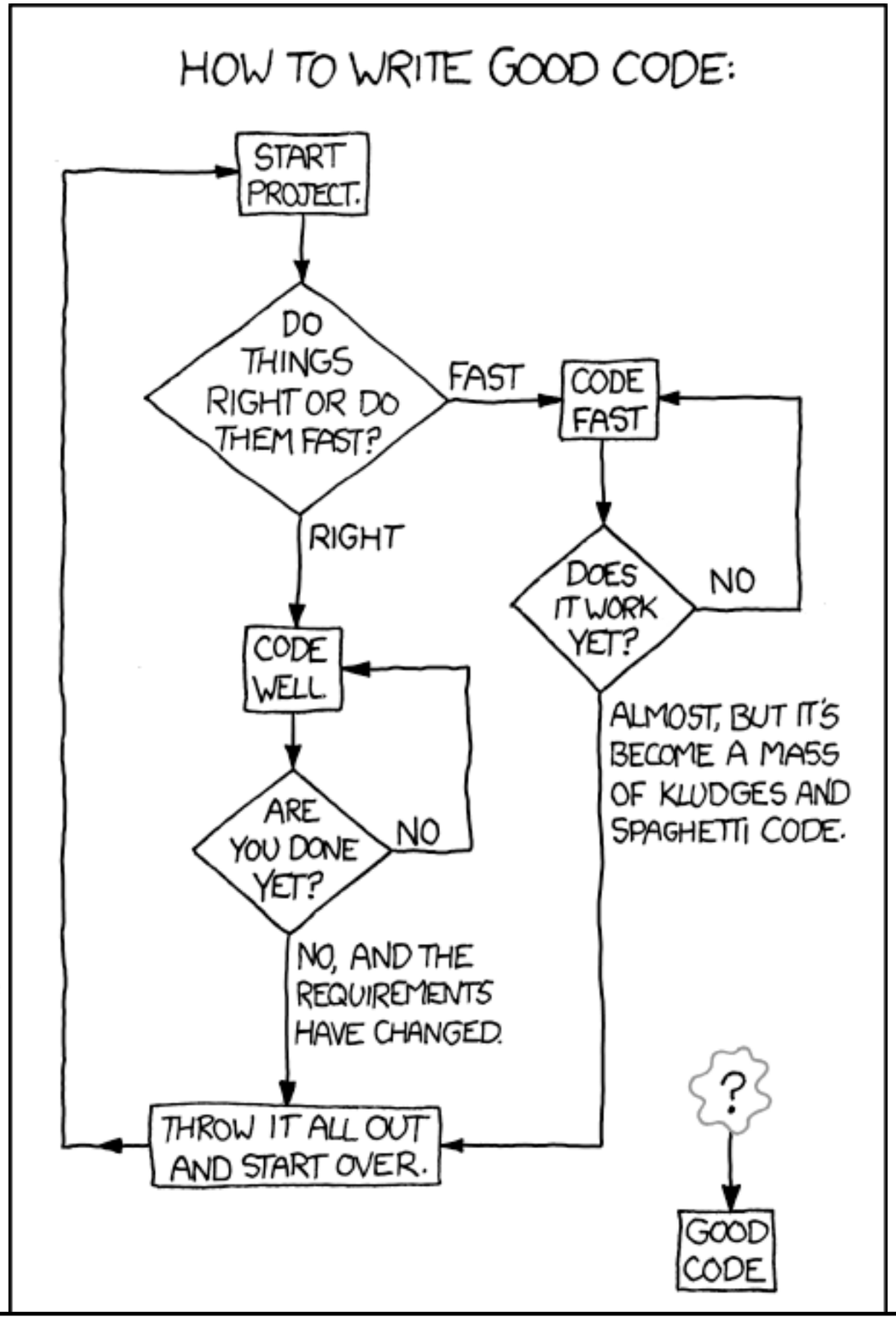
CSCI 4448/5448: Object-Oriented Analysis & Design
Lecture 01 — 08/28/2012

Two Announcements

- Ice Cream Social
  - The department will be hosting an Ice Cream Social as the first colloquium of the year:
    - **Thursday, September 6th, 3:30 PM to 4:30 PM in the DLC Conference Room**
      - This will be followed by the first CSUAC event of the year (free pizza)!
- CODEBREAKER
  - Free showing of an Alan Turing Documentary
  - **Math 100; Friday, September 7th, 6 PM (Doors open at 5:30 PM-ish)**
  - Documentary will hit theaters next year
  - Producer of film will conduct a Q&A after the film

This class teaches a style of software design that can help you reach the box labelled “Good Code”

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/>

About Me

- Associate Professor
  - Ph.D. at UC Irvine
  - 14 Years at CU;
    - Start of my 29th Semester!
- 11th time teaching this class
- Research Interests
  - Software & Web Engineering
  - Software Architecture
  - Hypermedia

Office Hours

- By appointment only
  - send me e-mail to schedule a meeting
- DLC 170M

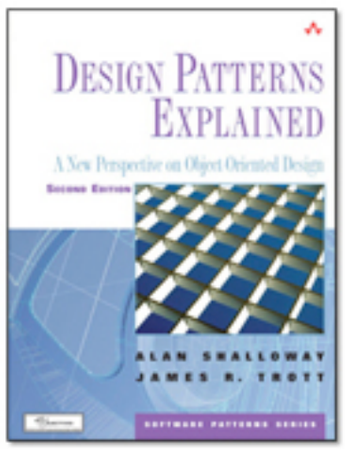
Class Website

http://www.cs.colorado.edu/~kena/classes/5448/f12/

Check the website every day! (I'm serious)

- To make it easy for you to track updates
  - Go to the "What's New" page and
  - Subscribe to the RSS feed
- Feed readers are available for all platforms
  - NetNewsWire, Reeder, FeedDemon, Google Reader, etc.
- 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 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** learn your name because I'll be taking attendance
  - this will be slow at first, but will get faster throughout the semester
- 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

- You will also gain experience with Android and iOS programming

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, October 16, 2012**
- Weeks 9 - 10: Intermediate and Advanced Android and iOS
- Weeks 11 - 13: Chapters 12-25 of the Textbook
  - Fall Break: November 19–23, 2012 (falls between week 12 and week 13)
- Weeks 14 - 15: Object Relational Mappings (Hibernate); Dependency Injection (Spring); Project Presentations

Course Evaluation

- Undergraduates
  - Class Participation & Attendance (10%)
  - Quizzes (10%)
  - Midterm (20%)
  - Homeworks (60%)
- Graduate Students
  - Class Participation & Attendance (10%)
  - Quizzes (10%)
  - Midterm (20%)
  - **Presentation (20%)**
  - Homeworks (40%)

CAETE students will not have attendance requirements but will have to meet class participation requirements (asking questions, contributing to on-line discussions, etc.)

Discussion

- Quizzes will be taken on the moodle
- Homeworks and the presentation will be submitted on the moodle as well
  - Some homeworks will be graded in class (peer review)
- Midterm
  - CAETE students will need to work with CAETE to identify a person to proctor their midterm exam; You will have from October 16th to October 23rd (one week) to take your exam and have it sent to me by your proctor

Honor Code

- You are allowed to work together in teams of 2 to 4 people on
  - the homeworks
  - the presentation (limited to 2 people)
  - the project (which is part of the homeworks)
- The quizzes and the midterm are individual work

- 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 submitted late incur a 15% penalty
  - You may submit a homework assignment and the presentation up to **one week late**
    - after that the submission will not be graded and you'll receive 0 points for it
- The quizzes, the midterm, and the final homework **may not be submitted late**
  - If you discover that you cannot attend the midterm on October 16th, you need to get in touch with me ASAP **before** the midterm to make other arrangements
    - trying to make arrangements **after** the midterm will not go well
- **To emphasize: if you miss a quiz or the midterm, you're out of luck**

Homeworks

- I'm going to take a different approach with some homeworks this semester
  - I'm going to have **you** participate in grading them during class
  - On the day a homework assignment is due, you will bring a hard copy of your assignment to class; the hardcopies will be distributed amongst your peers
    - if you work in a team, bring only one hard copy for the team
    - if you work in a team, you will grade as a team
- I will provide a grading rubric
- I will review the answers to the assignment
- You will assign a preliminary grade to the assignment
- The graded assignments will then be handed in and the grader will review them and assign final grades

Discussion on Plans for Homeworks

- Please be flexible and open to making this work
    - My goal in doing this is to increase your own knowledge of the course material
- CAETE students will do this at home for their peers
    - You'll submit assignments to the moodle and I'll redistribute them
    - You'll grade at home after you watch the lecture and then send the assignment with your preliminary grade to the grader
- The grader is involved in the process to guard against students who are either too harsh or too lenient in their grading

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/5828/f12/syllabus-statements.html> for more details

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

Discussion (I)

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

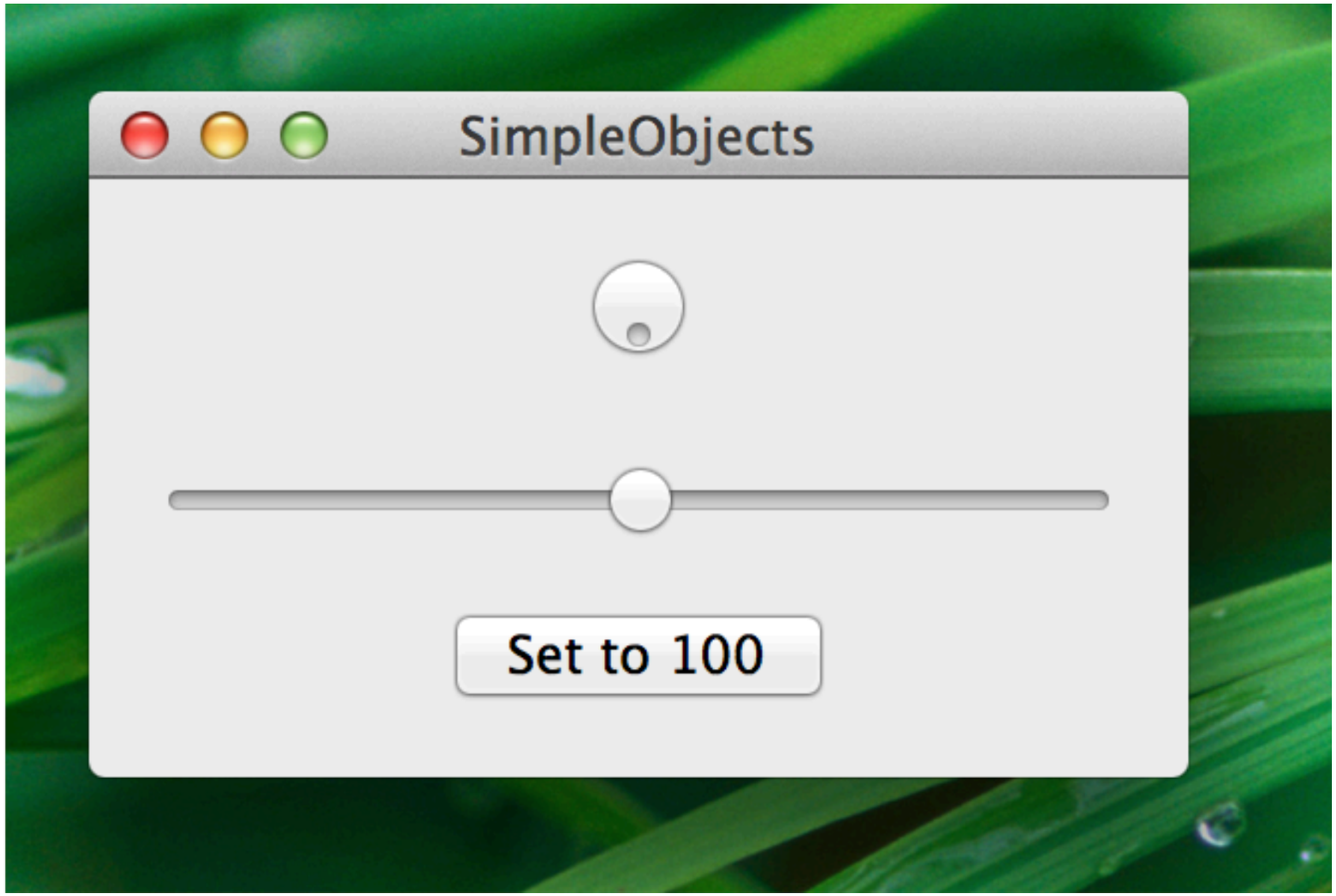
- When you create a program from scratch:
  - do you use OO techniques?
  - OO design heuristics?
  - design patterns?
- If not, what style of software design do you use?
  - What styles of software design are you aware of?

Discussion (IV)

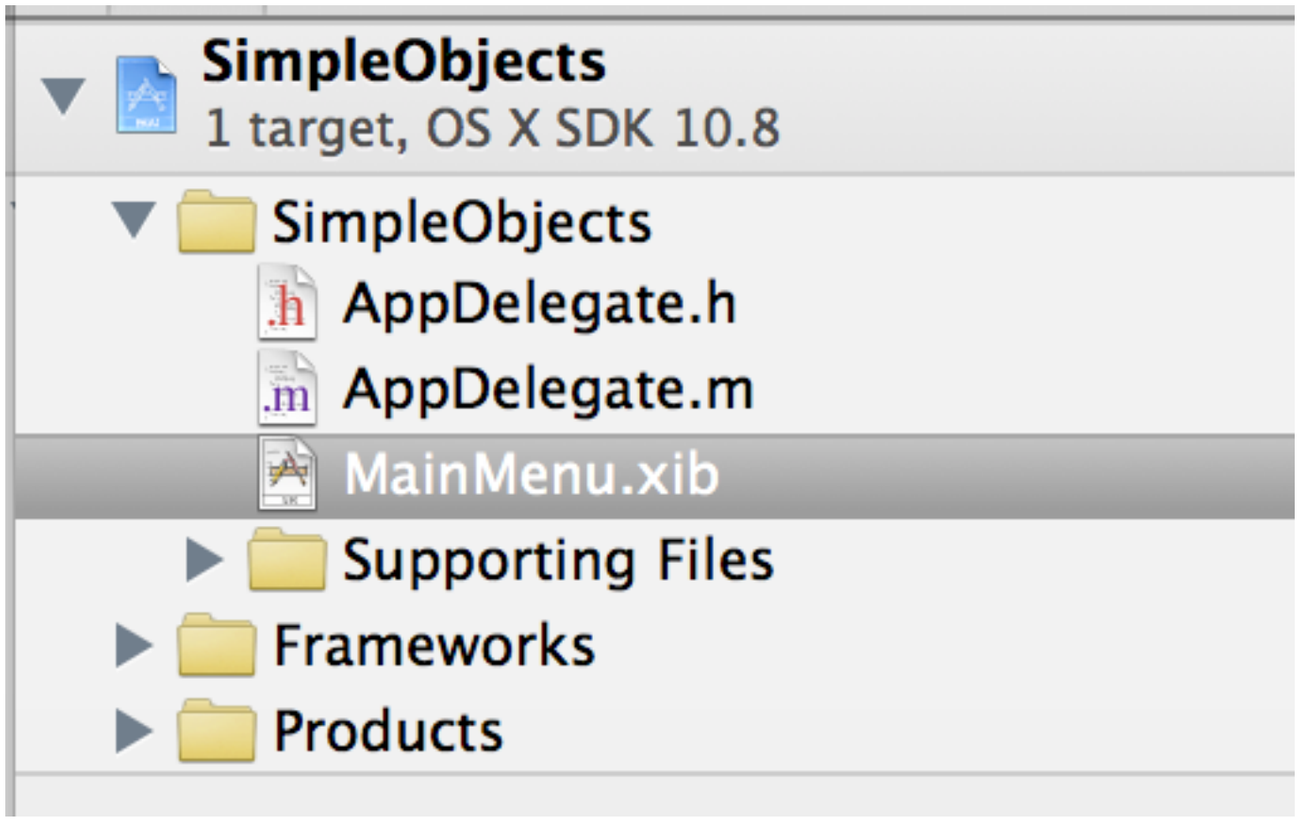
- What is design?
- What comes before design?
- What comes after design?
  - Do these questions make sense in software development?
- What would make the process of software design object-oriented?

Discussion (V)

How many objects do you think are working together to create the application shown on the right?

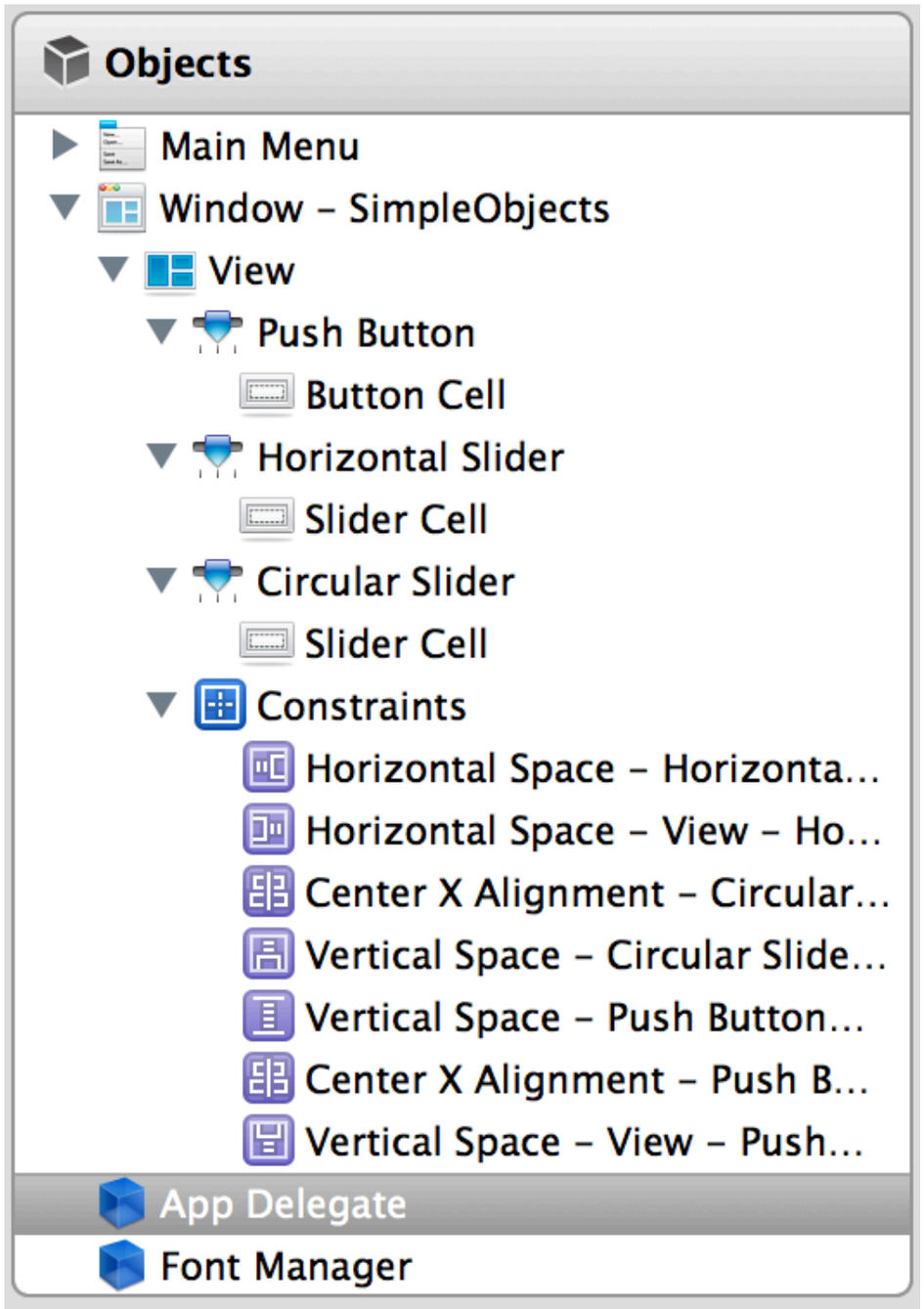


From a source code perspective... just one?



What's this .xib stuff?

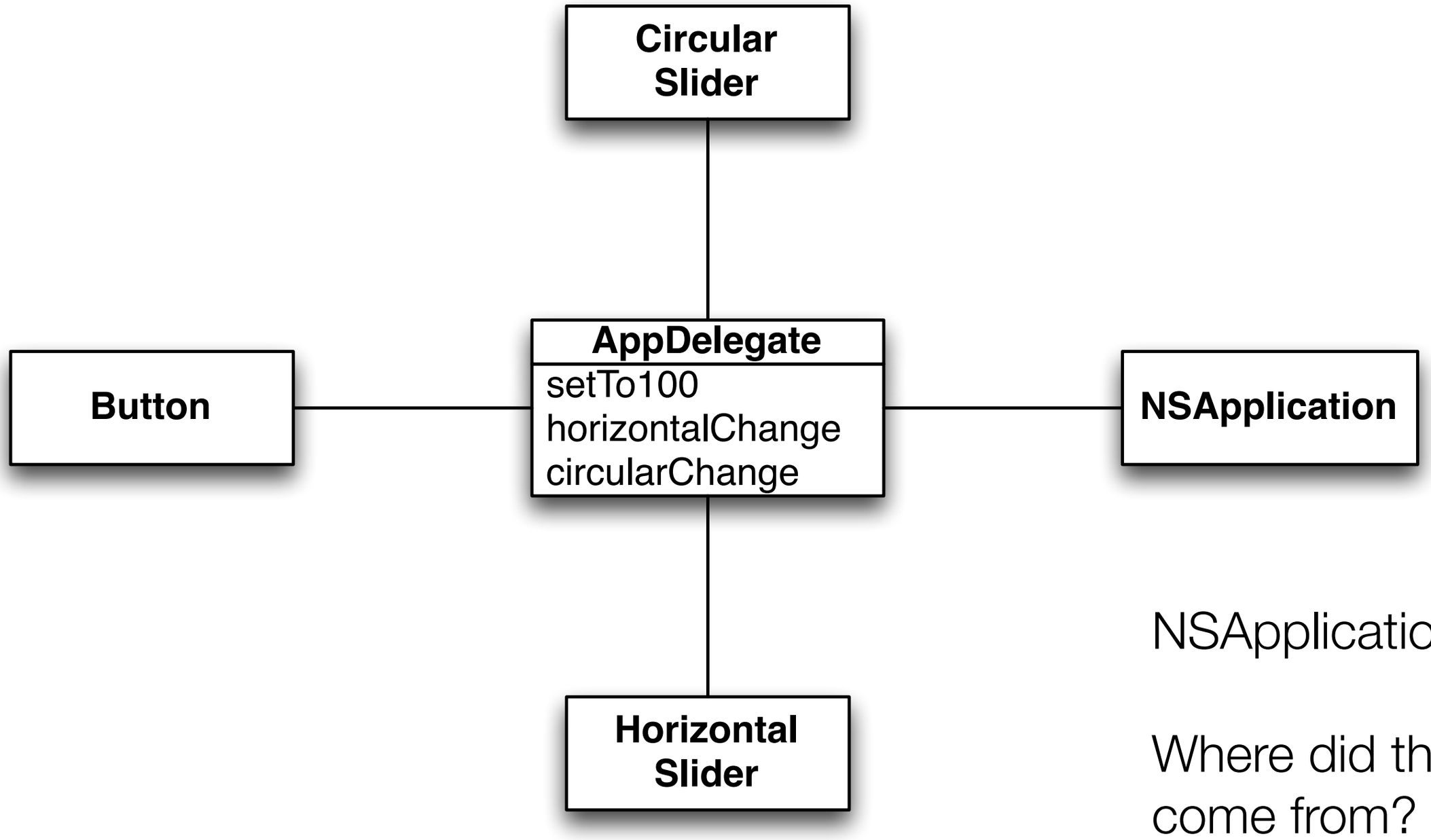
Oh, more objects... a lot more!



19 objects + AppDelegate = 20 objects

(Let's ignore what's hiding in the Main Menu object...)

To run, these objects connect and collaborate



NSApplication?

Where did that come from?

The bootstrap

- In Cocoa applications, there is a main.m file consisting of a C function
  - not object-oriented
- that looks like this
  - `int main(int argc, char *argv[]) {`
  - `    return NSApplicationMain(argc, (const char **)argv);`
  - `}`
- This is called the bootstrap; use of procedural code to create a single object
  - NSApplication
- that then loads the .xib file (creating all of the other objects) and starts processing events

Coming Up Next

- Lecture 2: The OO Paradigm
  - Read Chapter 1 of the Textbook
- Homework 1: To be assigned on Thursday