

Course Overview

CSCI 4448/5448: Object-Oriented Analysis & Design

Lecture 01 — 01/14/2019

Task number one

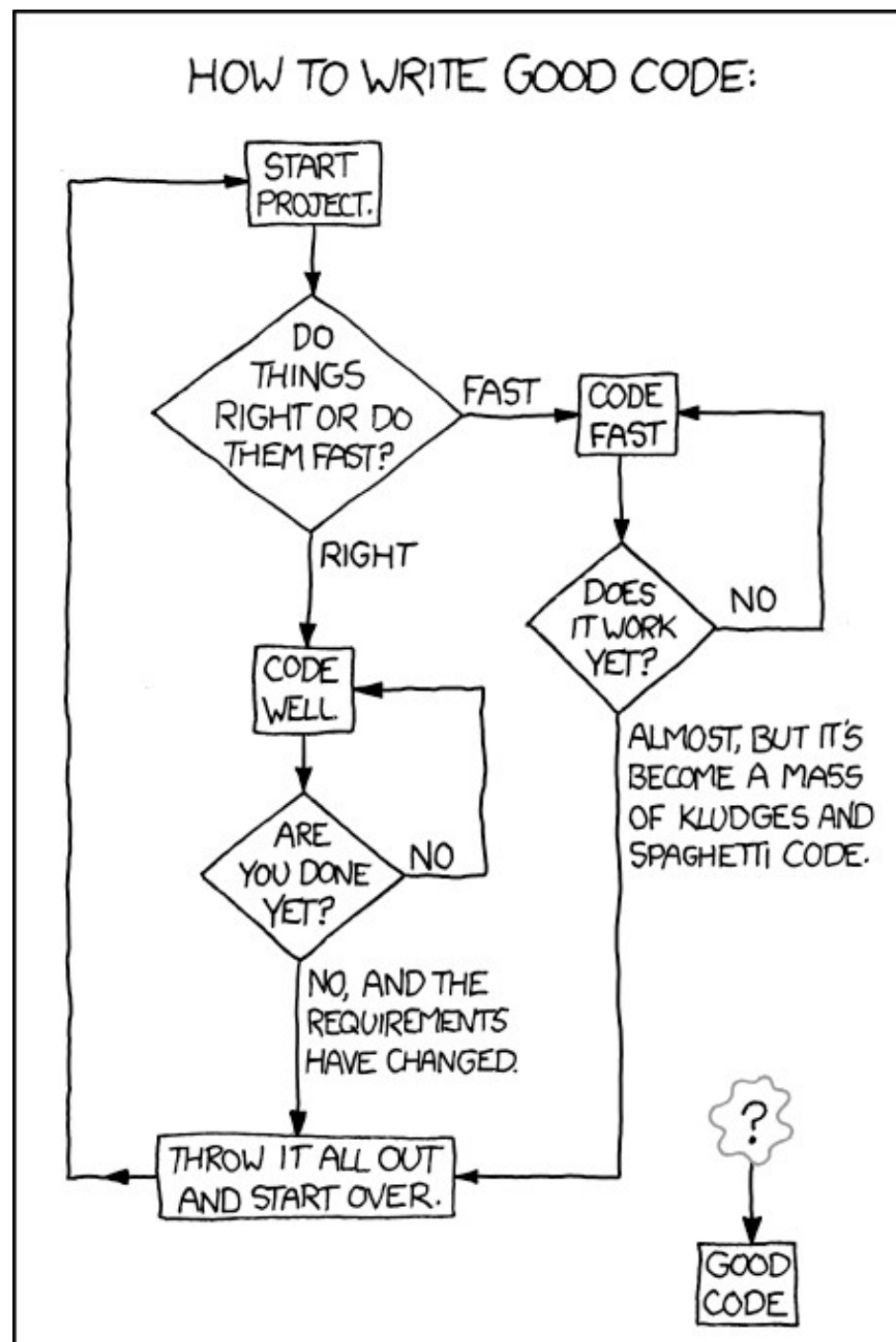
- Get a marker and a piece of card stock
- Fold it in half to make a little tent
- Write the name you'd like to be called on that
- Face the name towards me
- Try to remember to bring that to class for the next several weeks

Acknowledgement & Materials Copyright

- I'd like to start by acknowledging Dr. Ken Anderson
- Ken is a Professor of the Department of Computer Science and the Associate Dean for Education for the College of Engineering & Applied Science
- Ken taught this OOAD class on several occasions, and has graciously allowed me to use his copyrighted material for this instance of the class
- Although I will modify the materials to update and personalize this class, the original materials this class is based on are all copyrighted © Kenneth M. Anderson; the materials are used with his consent; and this use in no way challenges his copyright

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/>>

Bruce Montgomery, PhD, PMP

- Academic Background
 - PhD in Computer Information Systems from Nova Southeastern University in Ft. Lauderdale, FL - Sharks
 - Masters in Mathematics (Statistics and Computer Science focus) from Trenton State College in Ewing, NJ – Lions
 - Now The College of New Jersey
 - Bachelors of Science in Engineering (Computer Science) from the University of Florida, Gainesville, FL - Gators
 - PMI Certified Project Management Professional (PMP)
- 35+ Years in software development and engineering management
- Current full-time position: Vice President of Engineering at Inovonics in Louisville, Colorado
 - Inovonics makes wireless IoT (Internet-of-Things) devices and related web/cloud software for multiple markets
- Proud grandfather (say Hi to Wyatt and Emily)
- Science-fiction and comics fan (surprise!)



Something You Need to Know

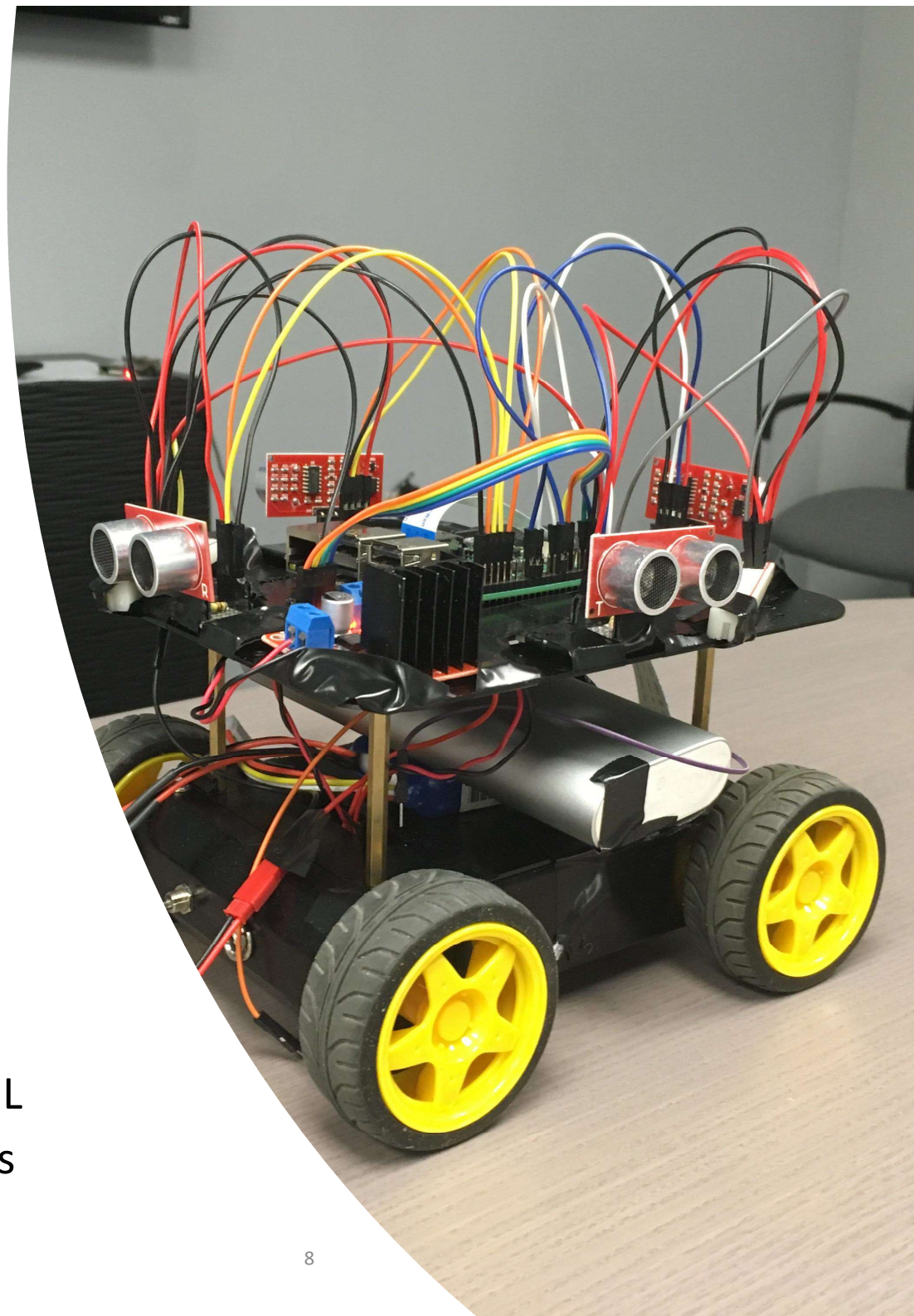
- I am hard of hearing and I use hearing aids
 - Is this from chronic ear problems, genetics, or standing in front of very loud Hartke double-stacked bass amplifiers? The answer is yes.
- I want you to ask questions and I want to answer your questions, but you'll have to be patient with me if I don't get it the first time
- Please don't be offended if I walk over to you to try the question again
- You must use your loudest indoor voice (nearly outdoor level), quiet whispers will never work, even if you're standing near me. If we need to set up a conversation before or after class to get past this, we certainly can
- So, I apologize in advance for any difficulties in this regard
- Just realize, if you ask what $2 + 3$ is and I say Chicago, this is 99% of the time me thinking I heard you and trying to respond
 - The other 1% of the time, I'm just wrong about $2 + 3$

Me at CU-Boulder

- I have been an Adjunct Lecturer in the ECEE department for the graduate embedded systems engineering (ESE) program
- I have developed and taught a class called Embedded Interface Design three times, in Fall and Spring of 2017, and Fall 2018
 - It is moving from a special topics class to a standard ESE elective, and I am committed to teach it again in Fall 2019
 - I'm also working now on a Coursera MOOC version of the class for the online ESE Masters Degree program
- This is my first course for the Computer Science department
- I'm interested in usability and user experience, IoT and Cloud architectures, software engineering practices, data analysis and simulation, computer graphics, and recently cybersecurity
- Details about me at: <http://ecee.colorado.edu/~brmo3998/>

Embedded Interface Design for ECEE

- Three main focus areas
 - M2M and IoT Protocols – Interfaces to Things
 - User Experience and Interface Design for Embedded Devices – Interfaces to Humans
 - Rapid Prototyping for Embedded Devices – leveraging Single-Board Computers (SBCs), Microprocessors, Embedded Linux, and Cloud/IoT Tools (ex. AWS IoT Framework)
- EID class uses Raspberry Pi, Linux, AWS, Python, Node.JS, QT, HTML/CSS/Jquery, other platforms and tools – very hands on project-oriented
- Includes study of Use Cases and UML
- Pictured is one of the student team's notorious super-projects



Other Class Staff & Class Schedule

- Teaching Assistant: Manjunath Rao Nagaraja Rao (Manjunath.NagarajaRao@colorado.edu)
- Graders: Neethi Shetty (Neethi.Shetty@colorado.edu) and Kumar Bhargav Srinivasan (Kumar.Srinivasan@colorado.edu)
- Class Schedule: Monday/Wednesday/Friday, 11:00 AM – 11:50 AM, Classroom ECCS 1B12
- Generally, Bruce will lecture on Mondays and Fridays
 - I often travel during the week – if I will be unavailable for a class, I will announce it on the class Canvas site, Mondays and Fridays are usually OK
- Wednesdays (starting 1/23) Manjunath will answer questions on quizzes, homeworks, projects etc. in an open recitation session, Bruce will not attend these sessions

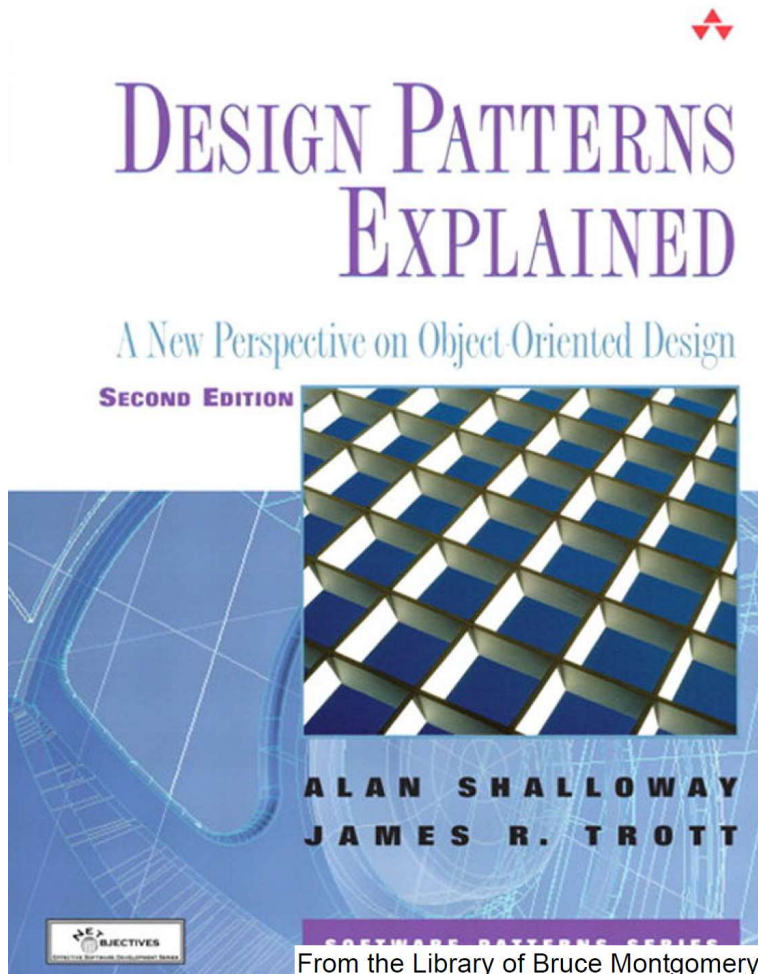
Office Hours & Communications

- I'm not on campus, except for class on Monday and Friday
- Your first and best source for questions will be the TA and graders, if they don't have a good answer, they'll contact me
- If you need to reach me, your best bets are Slack or e-mail
 - My e-mail is bruce.r.montgomery@colorado.edu
 - If you forget the .r. you'll be e-mailing a professor in library science
- Sign up for the class Slack IM (mobile and web tools available)
 - Slack site is oadcuspring2019.slack.com
 - https://join.slack.com/t/oadcuspring2019/shared_invite/enQtNTIxOTQzMtYzOTg0LTAxNzYwYmM1ZGY4MGMwODAyN2Y0ZTQyMzNkYThhMzE3MzA4NmRIZWM0YTUyYTRIYzA5NzRjODQ1MTE1MTJlZTc
- I try to answer both as soon as I can, I usually have Slack open on every device I own
- If we need a longer meeting, I can always setup a Zoom web conference
- If all else fails I can set up a time to come over to campus
- Don't hesitate to ask questions or ask for help when you need it!

Class Website - Canvas

- I use Canvas for all announcements, assignments and class materials
- The class Canvas site is:
 - <https://canvas.colorado.edu/courses/24796>
- I will post lectures for class on the site just before (or in some cases, just after) each class in PowerPoint and PDF
- Please sign up for notification of announcements from the site!

Textbook



- Design Patterns Explained
 - A New Perspective on Object-Oriented Design, Second Edition
- Alan Shalloway and James R. Trott
- Addison Wesley, © 2005
- ISBN-13: 978-0-321-24714-8
- E-book available at: <https://www.pearson.com/us/higher-education/program/Shalloway-Design-Patterns-Explained-A-New-Perspective-on-Object-Oriented-Design-2nd-Edition/PGM213720.html>
- I did not have time to get the book into the campus bookstore, but it is available through on-line vendors
- 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

Other Resources

I've been interested in OOAD for a long time, especially UML, Patterns, and Object-Oriented design considerations. This is a chunk of my home bookshelves related to OOAD.

The “Gang of Four” Gamma, Helm, Johnson, and Vissides book on Design Patterns is the famous one, but if I was going to recommend one from this stack, it would be **Head First Design Patterns** by Freeman and Freeman. It has a very casual presentation, but it's a quick path into thinking about design with patterns – it'll sneak into our discussions, as will others of these.



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 go too fast or to back up and try again
- I **will** try to learn your names but...
 - This is not a skill I excel at
 - Please bring your name card with you for at least the first half of the semester
 - Forgive me if I have to ask
- Learning by Doing
 - I will try to insert in-class activities where I can, but sometimes, it'll be lecture-y
 - Quizzes will bring out the most important lecture points
 - 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 OO programming, hopefully with languages you're interested in learning

Course Structure (**Tentative**)

- Weeks 1 - 6: Chapters 1 - 11 of the Textbook
- Weeks 6 - 9: Introduction to some OO languages
 - I need to do some work on this part
 - I will certainly talk about OO in Python and in Java
 - From there, we'll see...
- Week 8: Midterm; Exact Date: **Monday, March 4, 2019**
- Weeks 9 - 11: Chapters 12-25 of the Textbook
 - Spring break, week of 3/25/19
- Weeks 12 - 13: Refactoring, Object Relational Mappings, Dependency Injection
- Weeks 14 – 15: Project Demonstrations, Graduate Presentations, Wrapup

Ken Anderson's Lecture Videos

- I have posted Dr. Anderson's videos on the Canvas site for the class
- Viewing these videos is not required, but certainly you can use them to supplement discussion of topics we cover in lectures
- I will let you know if I think a video posted there is particularly useful

Course Evaluation

- Undergraduates
 - Class Participation & Attendance (10%)
 - Quizzes (10%)
 - Midterm (20%)
 - **Homework/Semester Project (60%)**
- Graduate Students
 - Class Participation & Attendance (10%)
 - Quizzes (10%)
 - Midterm (20%)
 - **Graduate Presentation (20%)**
 - **Homework (40%)**
- CAETE/distance students will not have attendance requirements but will have to meet class participation requirements
 - show engagement - asking questions, contributing to on-line discussions, etc.

Assignments Discussion

- I will be taking attendance for lectures! It counts for your grade
- Quizzes will be taken on Canvas
 - I will drop your lowest quiz grade, and I will let you take quizzes twice, highest grade counts
- Homeworks and Presentation will be submitted on Canvas
 - Documents will ALWAYS be PDFs
 - Code will ALWAYS be a GitHub repository URL
- Midterm
 - CAETE/distance 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

GitHub use

- I am requiring use of GitHub to store source code and readme markdown files for projects
- Do you know GitHub?
- Do you know how to write Markdown files for documentation?
- I'll gather up some reference material for you, but I'm not planning to teach this
- Please use the Wednesday recitation with our TA to review any questions you have about using GitHub (or find me)

Honor Code

- You are allowed and encouraged to work together in teams of 2 to 4 people on
 - the homework assignments
 - the presentation (limited to 2 people)
 - the semester 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:
 - <https://www.colorado.edu/sccr/honor-code>
- I will pursue action through the student conduct office if need be. A cautionary tale...

Late Policy

- You may submit a homework assignment and the graduate presentation up to **one week late**
 - Assignments submitted late (after the Canvas due date and time) incur a 15% penalty
 - 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 or the final homework submission, you're out of luck – no exceptions.**

Syllabus Statements

- The University asks that various policies be presented to students at the start of each semester. These policies include
 - Disability Accommodations
 - Classroom Behavior
 - Honor Code
 - Discrimination and Harassment
 - Religious Observances
- See the Syllabus on the Canvas site for more details

Programming Languages

- Examples will be written in various languages: Java, Python, Objective C are most likely
- OO Programming is **NOT** a central topic of the class
 - The thought is that analysis and design are the hard parts of OO software development
 - However, I will be devoting some lectures to introduce some OO languages and frameworks that illustrate OO elements
- Assignments
 - Note: You will be required to write **some** homework assignments in the Java language, otherwise any OO language may be used

Language Bias?

- The materials Dr. Anderson provided include iOS and Android development in Java and Objective-C
- My own background was initially Microsoft-focused, but I have shifted to Linux-based development over time, I have no experience with native iOS or Android development
- I have experience with FORTRAN, Pascal, C, C++, Java, Visual Basic 6, VB.NET, C#, and Mathematica – often for GUI or analysis apps
- Most recently my development has been with Python, Matlab/Octave, R, and a little Java, but I don't spend as much time coding as I'd like
- So, I am still thinking about the language part of the lecture schedule – I can guarantee Python and Java will be there
- However, I am open to any OO Language you'd like to use or study in student presentations or projects

TIOBE Programming Language Index

- Programming language popularity rating
- <https://www.tiobe.com/tiobe-index/>

Jan-19	Programming Language
1	Java
2	C
3	Python
4	C++
5	Visual Basic .NET
6	JavaScript
7	C#
8	PHP
9	SQL
10	Objective-C
11	MATLAB
12	R
13	Perl
14	Assembly language
15	Swift
16	Go
17	Delphi/Object Pascal
18	Ruby
19	PL/SQL
20	Visual Basic

Discussion (I)

- How many people have used an object-oriented programming language before?
 - Java? Python? C#? C++? Objective-C? Ruby? Others?
- What features make a language [object-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?
 - a cloud-based application?
 - an embedded device?

Discussion (III)

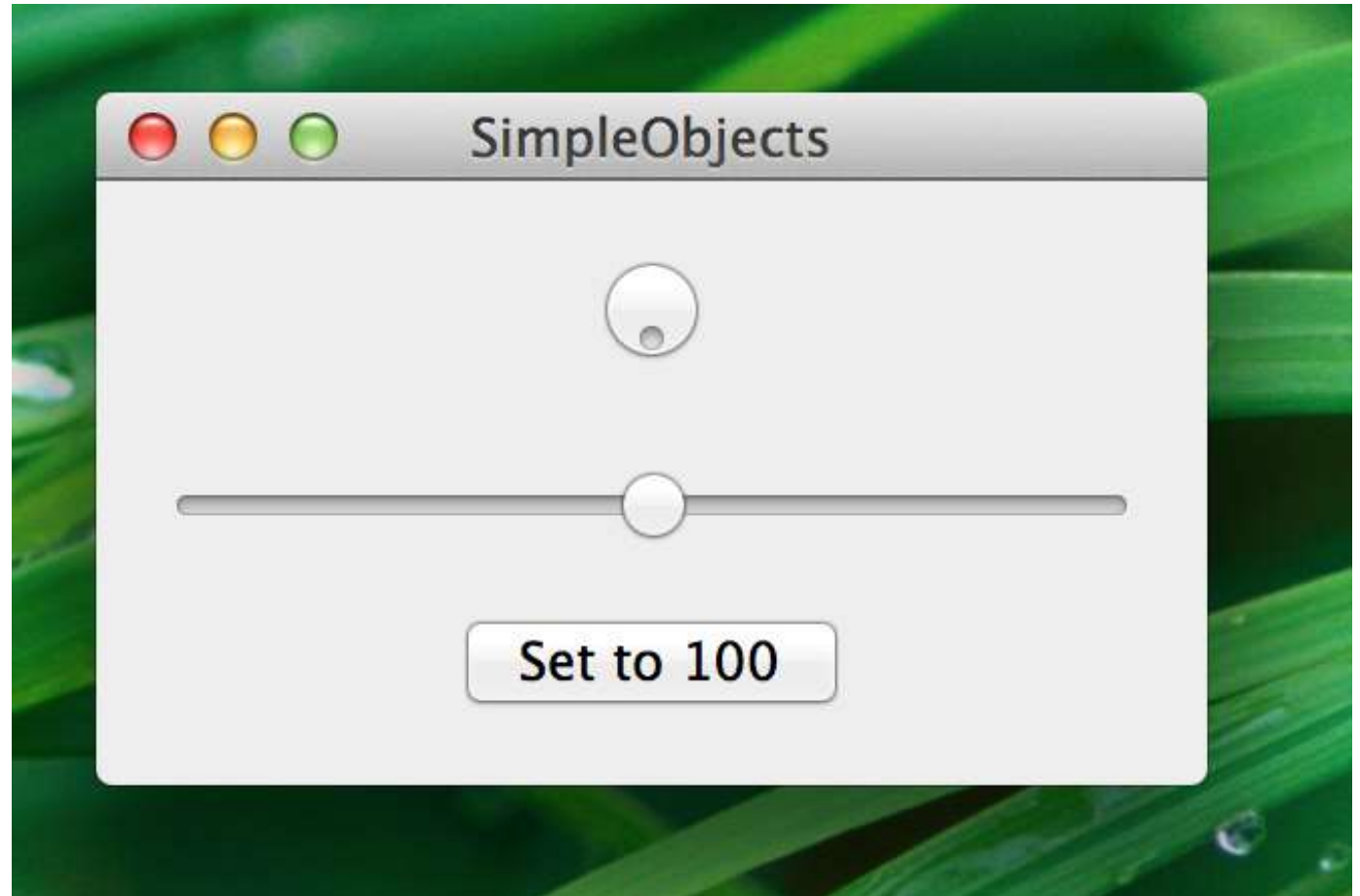
- When you create a program from scratch:
 - do you use OO techniques?
 - OO design heuristics?
 - design patterns?
- If not, what style of software design do you use?
 - What styles of software design are you aware of?

Discussion (IV)

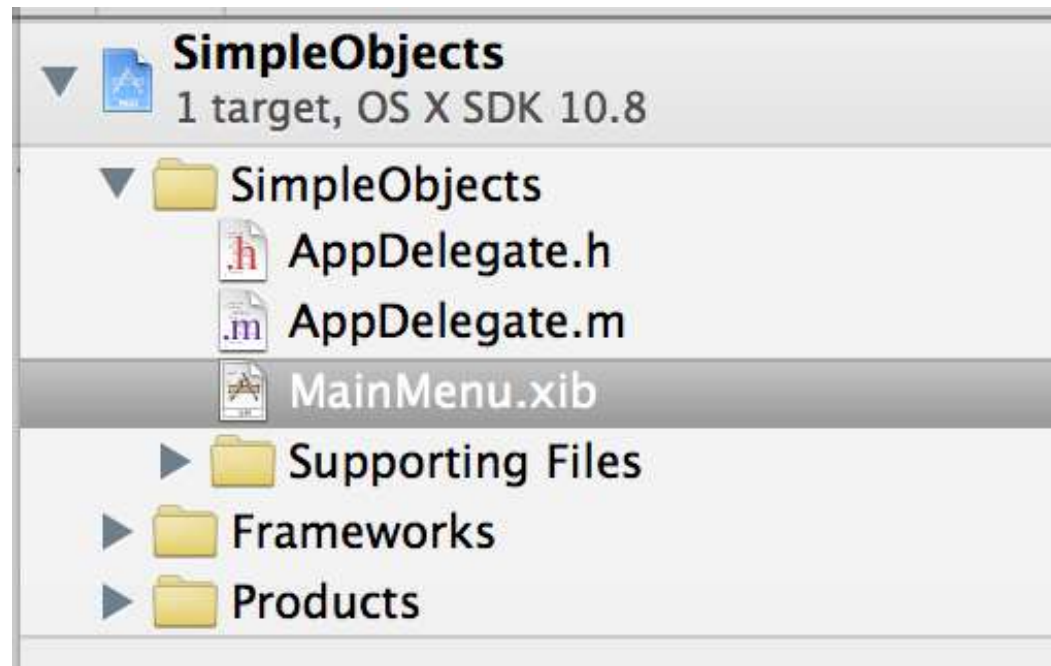
- What is design?
- What comes before design?
- What comes after design?
 - Do these questions make sense in software development?
- What would make the process of software design object-oriented?

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?

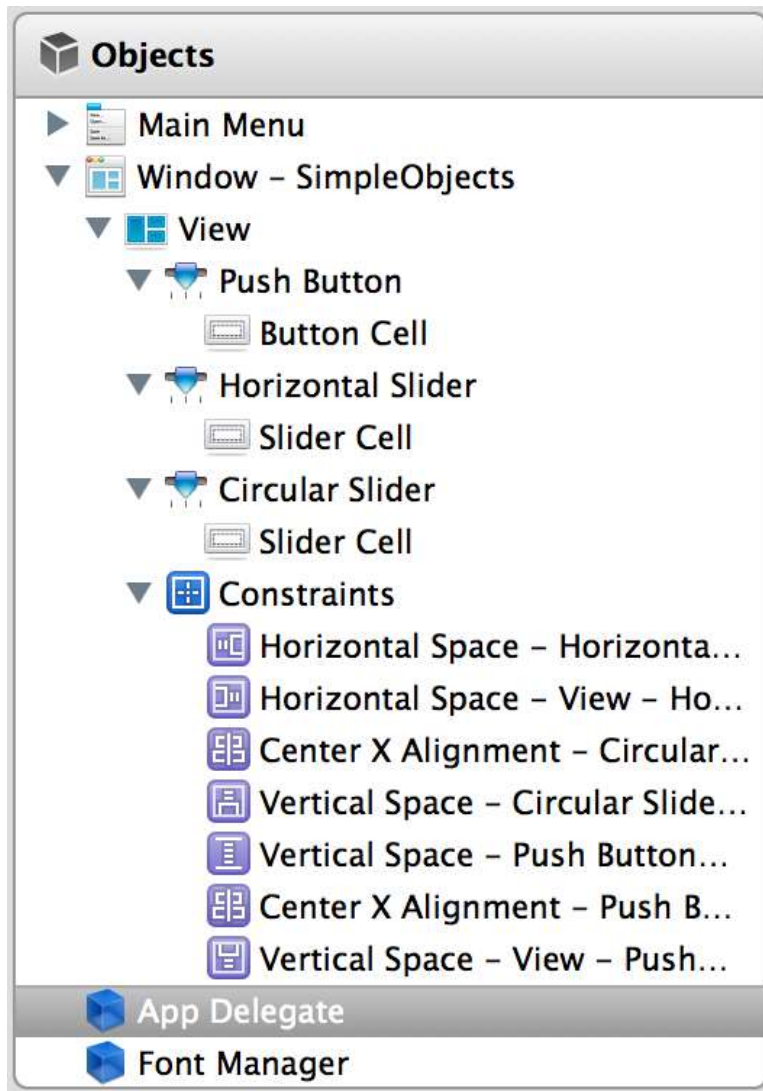


What's this .xib stuff?

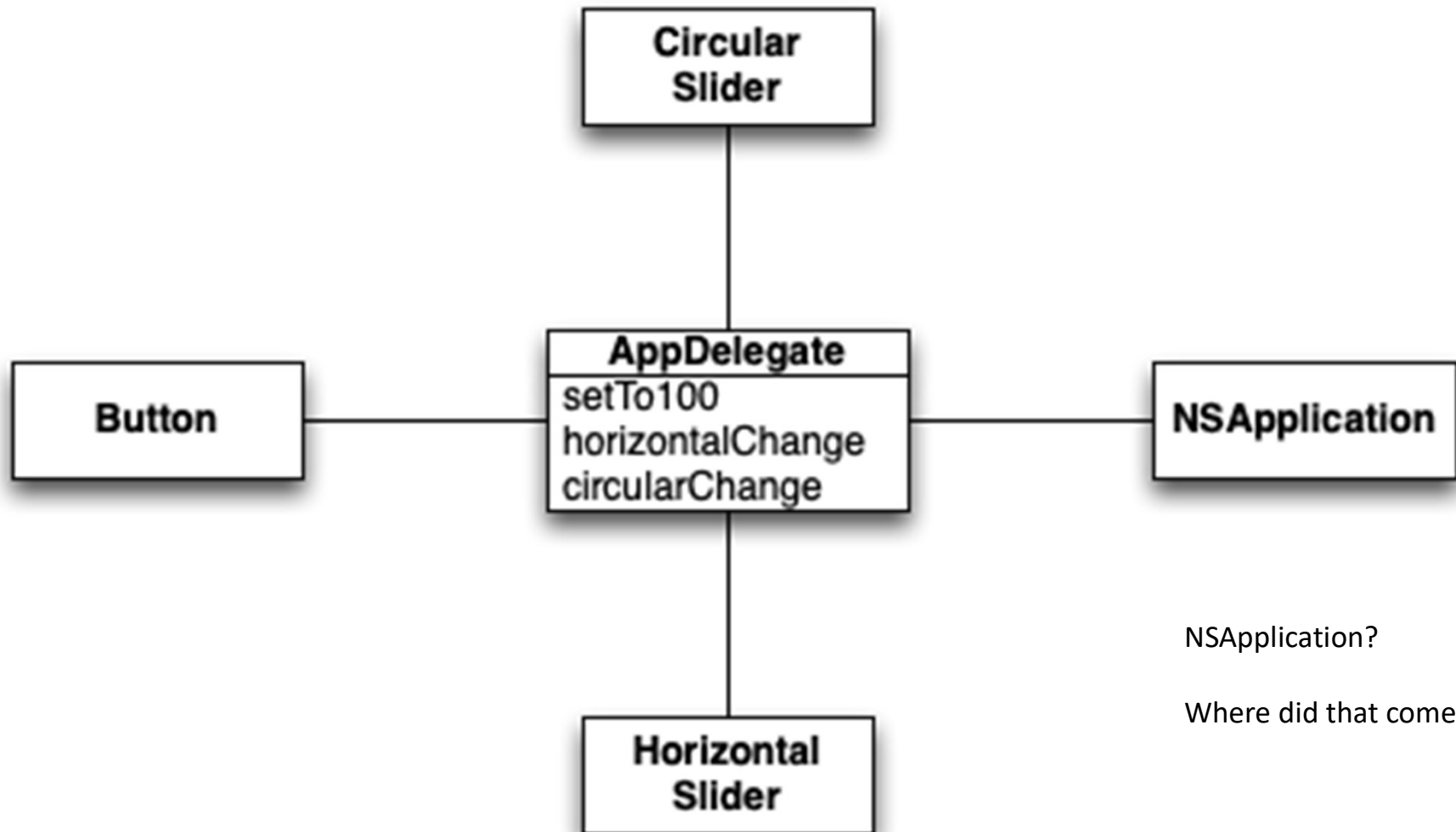
Oh, more objects... a lot more!

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

- Cocoa is Apple's native object-oriented application programming interface (API) for its desktop operating system macOS (per [Wikipedia](#))
- 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

Next Steps

- Get a copy of the textbook for the class
- Wednesday 1/16: No class, recitations start the following Wednesday
- Friday 1/18 - Lecture 2: The OO Paradigm
 - Read Chapter 1 of the Textbook before class
- Homework 1: Assigned on Friday
- Monday 1/21: No class, holiday
- Wednesday 1/23: Recitation period – good time to ask questions on the readings, GitHub, or on the homework assignment
- Friday 1/25: UML & OO Fundamentals
 - Read Chapter 2 of the Textbook before class