Object-Oriented Programming and User Interfaces

COSC346

Instructors

- Lecturer: Lech Szymanski, Owheo 2.49
  lechszym@cs.otago.ac.nz
- Lecturer: David Eyers, Owheo 1.25
  dme@cs.otago.ac.nz

Schedule

- Lectures:
  - Tuesday 13:00 – 13:50, Arts Building, BURN7
  - Thursday 13:00 – 13:50, Geology Building, QUAD4
- Labs (separate streams):
  - Wednesday 10:00 – 11:50, Owheo G.38 (Lab F)
  - Wednesday 12:00 – 13:50, Owheo G.38 (Lab F)
- Tutorials (separate streams):
  - Tuesday 10:00 – 10:50, Teaching College, T201
  - Friday 10:00 – 10:50, Drama Centre, DC102

NOTE:

- There will be **no tutorial first week**
- There is a **lab in the first week**

Reading

- The Swift Programming Language (2016), Apple Inc.
- C. Eidhof, A. Velocity (2016), Advanced Swift, Objc.io.
- A. Hillegass, A. Preble, N. Chandler (2015), *Cocoa Programming for Mac OS X (**5th ed**)*, Big Nerd Ranch Guides.
- Timothy Budd (2002), *Object-Oriented Programming (3rd ed)*, Addison-Wesley.
- Jenifer Tidwell (2006), Designing Interfaces, O'Reilly Media, Inc.

Grades

- Assignment 1: 20%, due Mon Sep 5th
- Assignment 2: 20%, due Fri Oct 7th
- Final Exam: 60%
- Your work must be your own.

Course Overview: Lectures

	Date	Title	Reading	Example code
1	Tuesday Jul 12th	Course overview		
2	Thursday Jul 14th	Introduction to Swift		
3	Tuesday Jul 19th	Classes, objects and methods		
4	Thursday Jul 21st	Working with objects		
5	Tuesday Jul 26th	Inheritance I		
6	Thursday Jul 28th	Inheritance II		
7	Tuesday Aug 2nd	Polymorphism		
8	Thursday Aug 4th	Memory management		
9	Tuesday Aug 9th	Object interconnections		
10	Thursday Aug 11th	Swift Libraries		
11	Tuesday Aug 16th	Object oriented design		
12	Thursday Aug 18th	Object oriented design patterns		
13	Tuesday Aug 23rd	OOP review		
14	Thursday Aug 25th	Introduction to application programming		
Study break				
Assignment 1 due, Monday, Sep 5th				
15	Tuesday Sep 6th	Application programming on the Mac		
16	Thursday Sep 8th	Model View Controller		
17	Tuesday Sep 13th	Cocoa: Windows and Views		
18	Thursday Sep 15th	Cocoa: Multiple windows		
19	Tuesday Sep 20th	Cocoa: Mouse and Keyboard Events		
20	Thursday Sep 22nd	Cocoa: Bindings		
21	Tuesday Sep 27th	Cocoa: Controllers and Undo		
22	Thursday Sep 29th	Cocoa: Preferences		
23	Tuesday Oct 4th	UI design		
24	Thursday Oct 6th	Usability and visual design		
Assignment 2 due, Friday, Oct 7th				
25	Tuesday Oct 11th	Guest lecture		
26	Thursday Oct 13th	UI review		

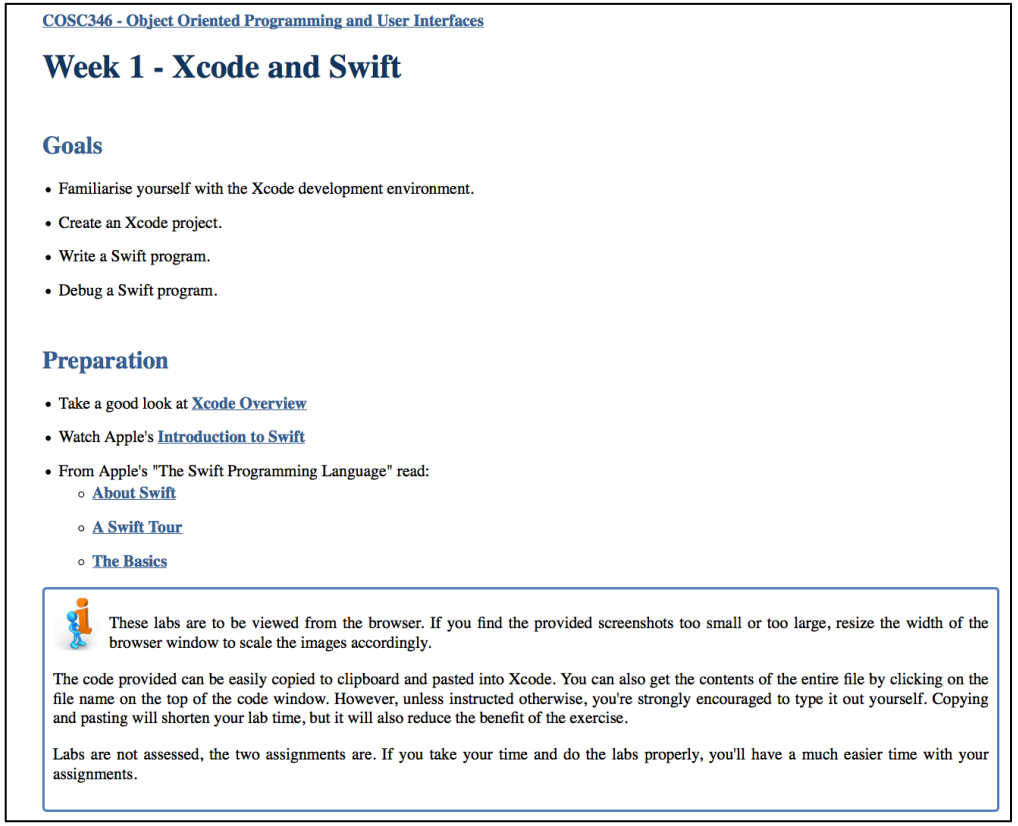
- Object Oriented Programming
  - General concepts: abstraction, encapsulation, inheritance, polymorphism, coupling, cohesion
  - Swift language and Foundation Framework
  - Swift development tools - Xcode
  - Object oriented design principles
- User Interfaces
  - Cocoa Environment and Xcode
  - Interface design principles: usability, basics of graphic design

OOP

Swift

Course Overview: Labs

- On the course webpage
- Not assessed
- First lab tomorrow



COSC346 - Object Oriented Programming and User Interfaces

Week 1 - Xcode and Swift

Goals

- Familiarise yourself with the Xcode development environment.
- Create an Xcode project.
- Write a Swift program.
- Debug a Swift program.

Preparation

- Take a good look at Xcode Overview
- Watch Apple's Introduction to Swift
- From Apple's "The Swift Programming Language" read:
  - About Swift
  - A Swift Tour
  - The Basics

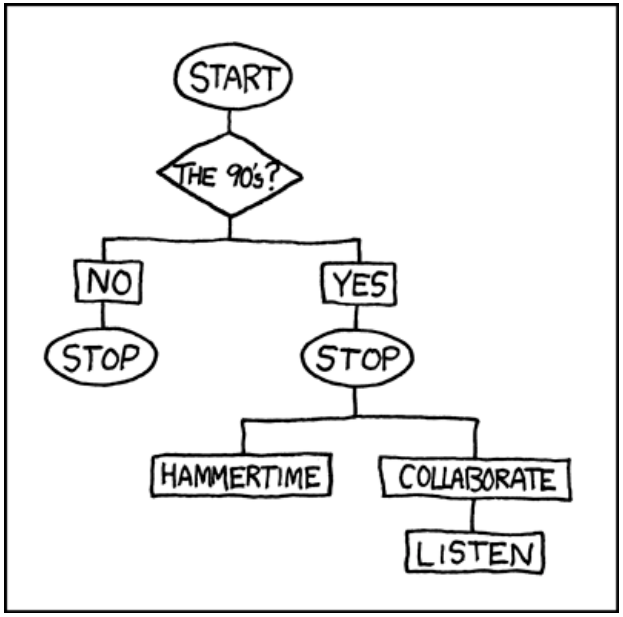
These labs are to be viewed from the browser. If you find the provided screenshots too small or too large, resize the width of the browser window to scale the images accordingly.

The code provided can be easily copied to clipboard and pasted into Xcode. You can also get the contents of the entire file by clicking on the file name on the top of the code window. However, unless instructed otherwise, you're strongly encouraged to type it out yourself. Copying and pasting will shorten your lab time, but it will also reduce the benefit of the exercise.

Labs are not assessed, the two assignments are. If you take your time and do the labs properly, you'll have a much easier time with your assignments.

What is OOP?

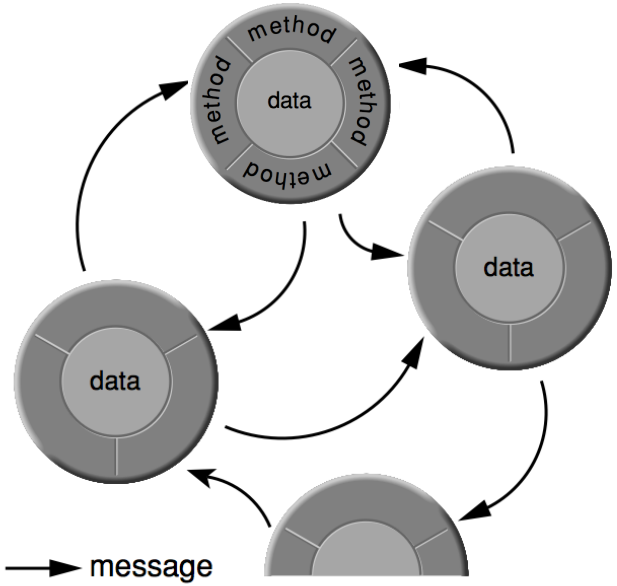
Procedural



xkcd.com

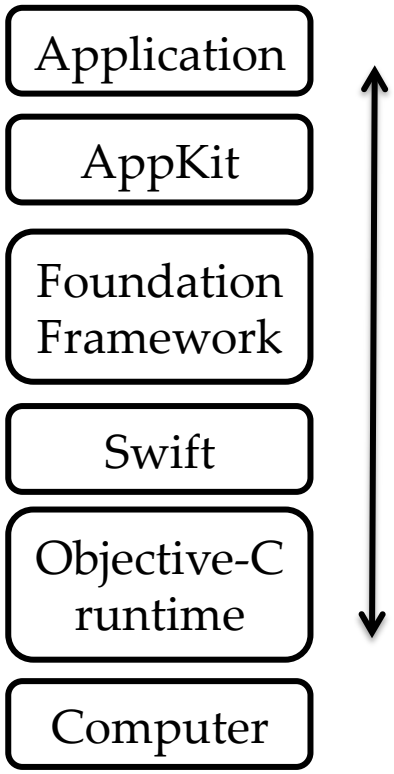
1. Functions act on data.
2. A program organises function calls to manipulate data.

Object-Oriented



1. *Objects* contain *encapsulated* data and associated *methods*.
2. A program describes how objects interact via *messages*.

Why OOP?



Speed	Code	Development	Environment	User Interface
Slow	**Re-usable**	**Team, Fast**	**Runtime Decisions**	**Complex, Graphical**
Fast	**Specific**	**Individual, Slow**	**Compile-time Decisions**	**Simple, Text-based**

Why Swift?

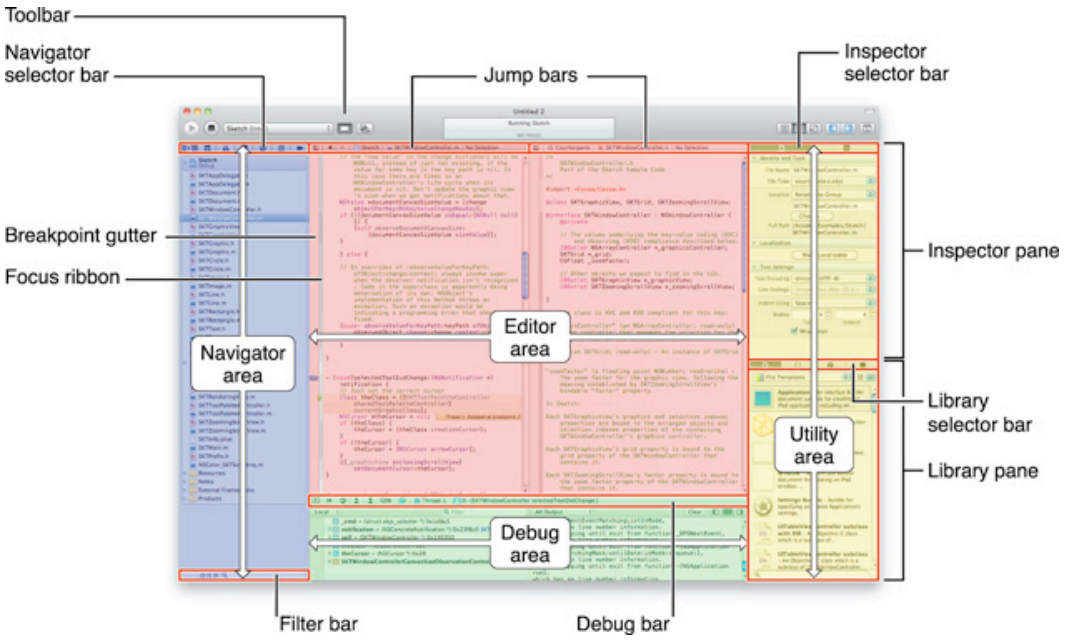
- Modern
  - Result of research on programming languages
  - Multi-paradigm – takes ideas from many languages, incorporating their best features (in this course we will focus on the Object-Oriented aspect)
- Safe
  - Compiler forces you to do things right
  - Emphasis on detecting errors at compile time rather than run-time
- Concise
  - Easier and faster to develop software
  - Easier to create development tools
- Cocoa environment – good example of natural progression from OOP to User Interfaces

What is Cocoa?

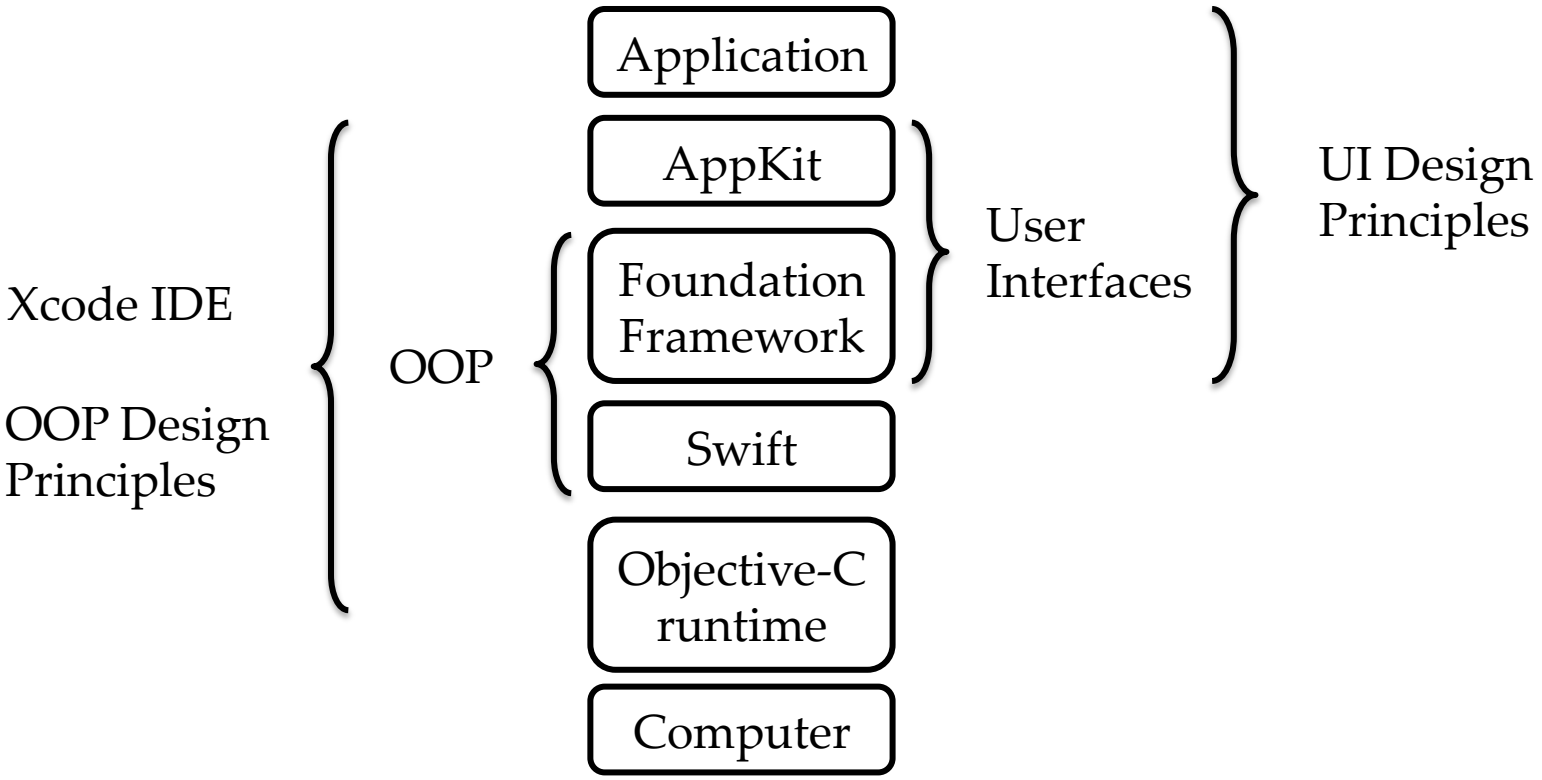
- Object-oriented framework for application development for OS X and iOS
  - In this course we will focus on OS X only
- "Its elegant and powerful design is ideally suited for the rapid development of software" – Cocoa Fundamentals Guide (2010, retired), Apple Inc.
- Huge number of classes and frameworks
  - Overwhelming for the first-time user
  - Powerful environment that abstracts away a lot of the details of application programming – you can concentrate on high level functionality

What is Xcode?

- Integrated Development Environment (IDE) for application development for OS X and iOS
  - It comes with iOS platform simulator
- Compiler and debugging tools
- Cocoa libraries and frameworks
  - Interface builder – graphical interface for creation of user interfaces
- Editor and tools for analysis



Mac Platform



Goals

1. Object-Oriented Programming:
   a) Learn Swift language.
   b) Understand OOP design principles.
2. User Interfaces:
   a) Learn Application Kit Framework.
   b) Understand UI design principles.