

01 Mo Lab Notes

ITPG-GT 2372 - 1 Mobile App Development Lab

3:20 PM - 5:50 PM Friday

1/28/2022 - 5/6/2022

room 409 @ 370 Jay Street,

Room 409 Loc: Brooklyn Campus

One of the most transformative consumer products in history, the iPhone remains the standard bearer for great design and user experience. With the latest versions of iOS and iPhone, Apple puts depth sensing and augmented reality in our pockets. How do we take advantage of this incredible platform to produce our own compelling experiences?

This course will be a hands-on workshop where we explore the world beyond generic apps and push the boundaries of what's possible on iOS hardware. Each week, you'll be asked to complete a programming exercise meant to foster your understanding of iOS application development. We'll leverage existing open source libraries to quickly build out your app with features such as real time communication and cloud storage.

Full-time access to an iOS device and a Mac laptop computer running the latest operating system and development tools are required.

Prerequisite: Some programming experience (such as ICM) and willingness to learn Apple's Swift programming language.

We aim to create distributed instruments for computed expression.

Course overview

<https://github.com/mobilelabclass-itp/content>

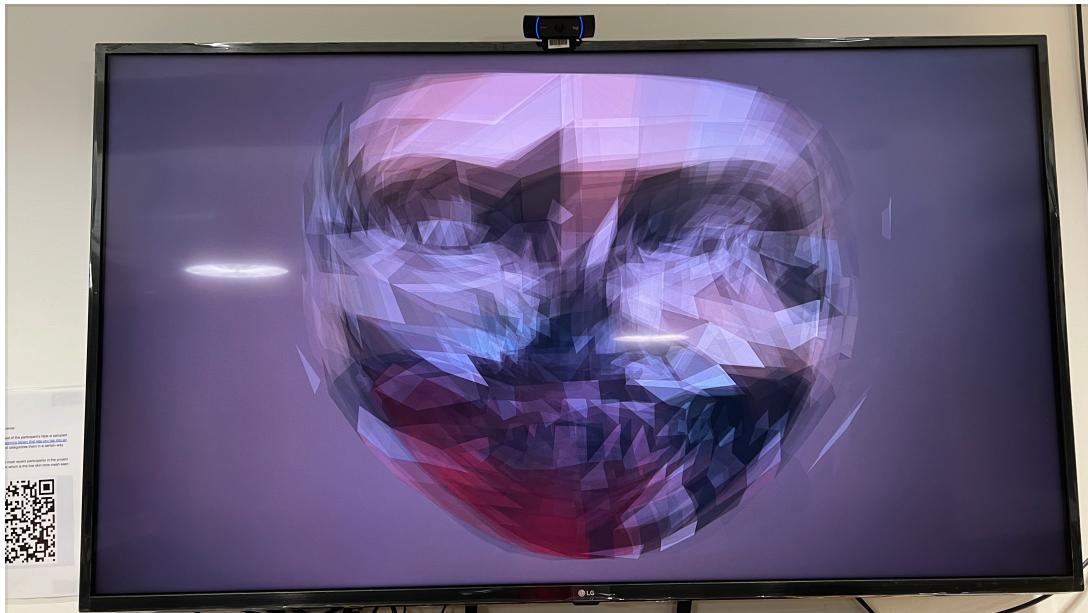
Grading:

- Regular Assignments 40%**
- Participation and Attendance 40%**
- Final Project 20%**

keep final project in mind as you level up on Swift

Class Intros

- [] create intro wiki page
- [] github user name reminder



Skin Tone screen, part of Colored Portraits installation

<https://jht1900.github.io/a/colored.html>

My DICE platform

2001 QuickDraw / QuickTime / C++ Carbon MacOS app

2005 Carbon / Javascript / PowerPlant

2017 Objective-C, AVFoundation

2021 NYU-ITP Installation - Javascript / p5js / ml5.js

I plan to update DICE to Swift / SwiftUI this year.

<http://www.johnhenrythompson.com/3-dice>

Example of evolving software ecosystem

Creative coding and creative learning

<https://www.youtube.com/watch?v=8QiPFmIMxFc>

Bret Victor Inventing on Principle

Jan 17, 2018

33 mins

<https://www.youtube.com/watch?v=ZfytHvgHybA>

Bret Victor - Stop Drawing Dead Fish

Aug 14, 2013

53:32

<https://www.youtube.com/watch?v=8pTEmbeENF4>

Bret Victor The Future of Programming

Jul 31, 2013

33 min

<https://dynamicland.org/>

<http://worrydream.com/#!/Showreel2012>

Bret Victor, beast of burden

Doug Engelbart

1968 NLS - mouse, video conference, multiple views of information

Alan Kay

Richard Stallman

Class discussion - your experience with creative coding.

Why Apple Swift?

Apple has demonstrated history of innovation

- 1984 Macintosh
- 2001 iPod
- 2007 iPhone

Goal to make computer personal expressive instruments

App for creative expression building on the unique palette (libraries) of the platform

-Break

The promise of Swift

<https://www.youtube.com/watch?v=MO7Ta0DvEWA>

Apple WWDC 2014 - Swift Introduction

10 min

Separating marketing speak from reality

Why Objective-C vs. Python, rather than C / C++?

Complexity and expressiveness is the underlying issue

Appropriate level of abstraction

In rapidly evolving software ecosystem finding good documentation and examples is hard

I could not find working code for demo of swift intro

Focus on SwiftUI to simplify learning

Apple migrating away from Storyboard/UIKit

Strategy

- Explore features of interest topic in WWDC videos, building towards a final project
- Organize your notes - I use VS Code
- Apple documentation
- Apple sample code
- Check resources. Current? Works?

Different approaches to learning - multiple paths

- Reading
- Watching

- Exercises
- Experimenting

Explore resources

https://github.com/mobilelabclass-itp/content/blob/main/weeks/01_intro.md#resources---apple-platform

<https://developer.apple.com/videos/all-videos>

WWDC Videos - What Apple says is possible on their platform
search to find area of interest: RealityKit, AVFoundation, Filters

>> Example WWDC treasure hunt:

<https://developer.apple.com/videos/topics/>

<https://developer.apple.com/videos/photos-camera>

<https://developer.apple.com/videos/play/wwdc2018/503/>

Creating Photo and Video Effects Using Depth

https://developer.apple.com/documentation/avfoundation/cameras_and_media_capture/capturing_photos_with_depth

https://developer.apple.com/documentation/avfoundation/cameras_and_media_capture/streaming_depth_data_from_the_truedepth_camera

>> Big stretch

https://developer.apple.com/documentation/realitykit/swiftstrike_creating_a_game_with_realitykit

SwiftStrike: Creating a Game with RealityKit

Create a multiplayer game with ARKit, RealityKit, and Swift using the SwiftStrike app as a guide.

>> Photos app curated memory:

Philadelphia Jun 24, 2016

I don't use social media

<https://github.com/jht1493/jht-site#jht-site>

How do we get there?

<https://docs.swift.org/swift-book/GuidedTour/GuidedTour.html>

- Apple Swift Docs - A Swift Tour

<https://www.linkedin.com/learning/swift-5-essential-training>

linkedin swift-5-essential-training

For new programmers

<https://www.raywenderlich.com/28797163-your-first-ios-swiftui-app-an-app-from-scratch>

Ray Wenderlich - your-first-ios-swiftui-app

In-depth multi week sessions

<https://www.hackingwithswift.com/100/swiftui>

www.hackingwithswift.com -- 100 days of swiftui

In-depth multi week sessions

<https://cs193p.sites.stanford.edu/>

Stanford University: CS193p - Developing Apps for iOS

For experienced programmer. First few sessions useful for beginners

<https://web.stanford.edu/class/cs193p/Spring2021/MemorizeL8.zip>

Demo Code

<https://www.advancedswift.com/learn-swift-cpp/>

Learn Swift for C++ Developers

Would be useful to extend with Swift vs. Javascript

<https://itp.nyu.edu/classes/mobile-lab/>

prior course @ itp.nyu.edu

<https://github.com/mobilelabclass/content>

prior course @ github

prior course was based on Storyboard - no SwiftUI

Hands on:

- Create your repo
- Use Xcode to create a playground in your repo
- Update your repo
- Add link to repo on homework page

Review homework

https://github.com/mobilelabclass-itp/content/blob/main/weeks/01_intro.md#homework

Explore 00-Playground pages

Download via GitHub Desktop or from [link to zip file](#)

Build a pages for experimenting with language features as you learn them
Hands on example

Recap repos

- content
- content wiki
- 00-Playground

Big picture recap and context

The Evolution of Software ecosystems

>> Swift is rapidly evolving

| | | |
|-----------|------------|------------|
| Swift 5.5 | 2021-03-12 | 2021-09-20 |
| Swift 5.0 | 2018-09-25 | 2019-03-25 |
| Swift 4.0 | 2017-02-16 | 2017-09-19 |
| Swift 3.0 | 2016-05-06 | 2016-09-13 |
| Swift 2.2 | 2016-01-05 | 2016-03-21 |

>> SwiftUI is new - intro 2019

Other Software ecosystems for mobile apps

Flutter/Dart

React Native

HTML5/Javascript/CSS/Electron (for the desktop)

Android Java/Kotlin

Where do they differ?

Where do they converge?

Compiled vs. Interpreted

Complexity

State management

Async

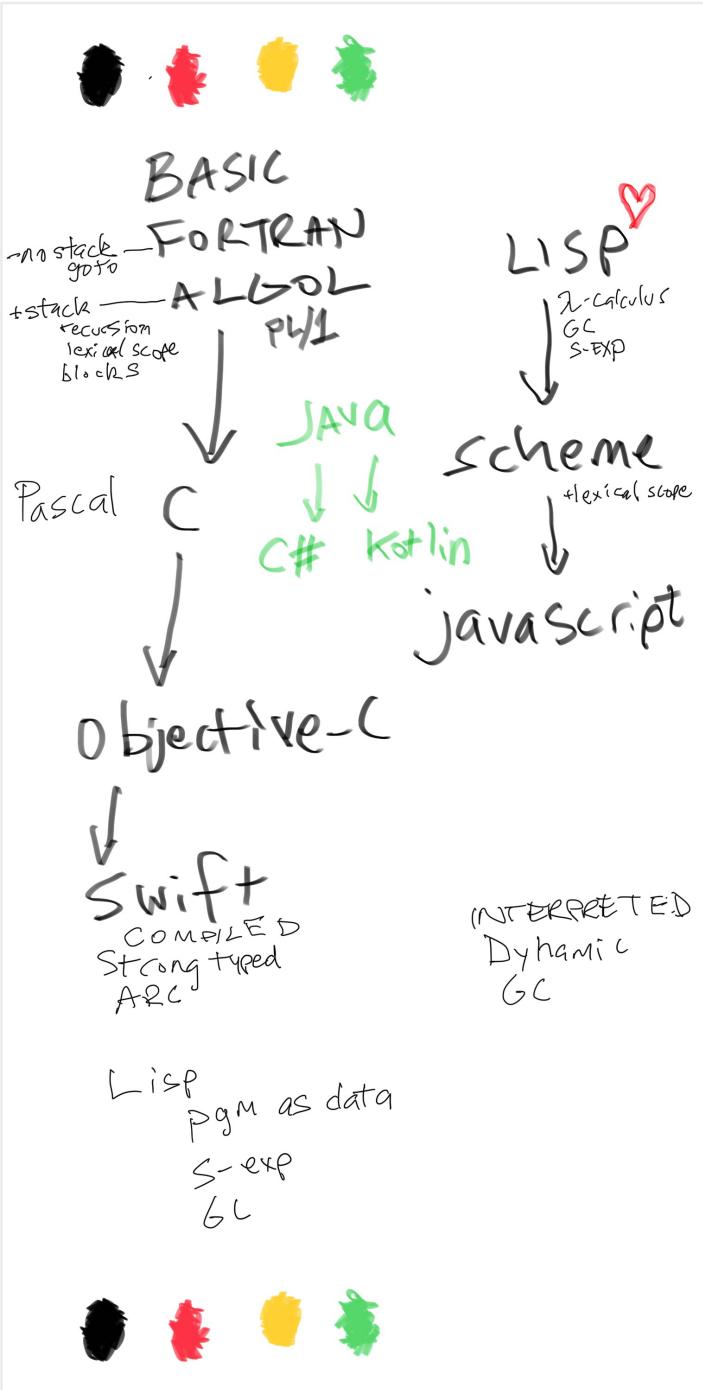
History of computer language evolution

inflection points

new programming paradigms

- BASIC

- Structured Programming
Algol lexical scoping vs. Fortran flat
Recursion
- Data Abstraction
--> Algol, C, Pascal
<-- Fortran, Assembler
- Dynamic
Lisp -> Scheme -> JavaScript
- OOP
Smalltalk
Object Pascal
Objective-C
Java
C# (Microsoft's response to Java)



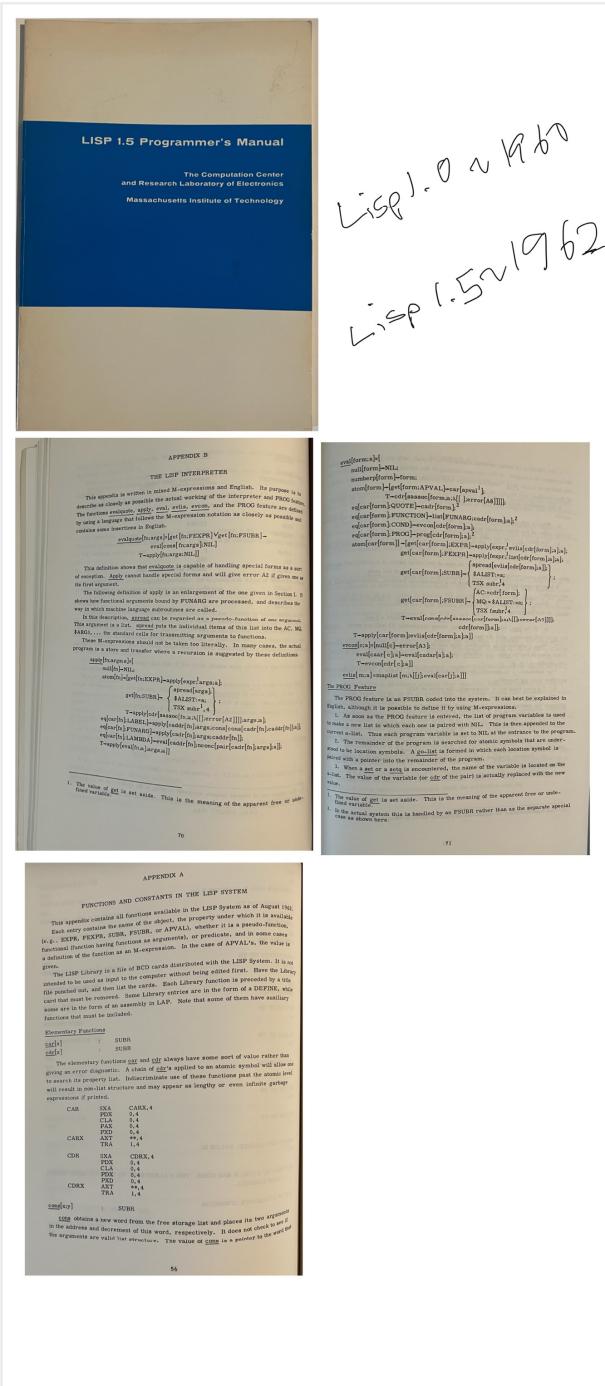
Lisp 1.5 Programmer's Manual

approx. 106 pages

Beautifully concise and expressive language - the root of Scheme/Javascript, Java/C# and other recent languages

Inspired me to write Lisp interpreter for PDP-8 (~1978), and Lisp compile for IBM 360 (~1983)

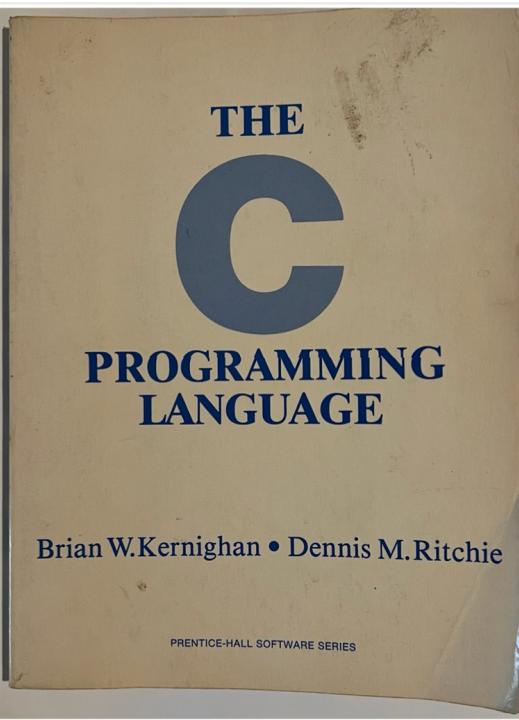
Foundation for my creation of the [Lingo scripting language](#) for Macromedia/Adobe Director (1990's)



The C Programming Language

approx. 227 pages

From complex to simple (Multics to Unix) and back to complex (C++)



APPENDIX A

218 THE C PROGRAMMING LANGUAGE

```
statement:
compound-statement
expression ;
if ( expression ) statement
if ( expression ) statement else statement
while ( expression ) statement
do statement while ( expression ) ;
for ( expression-1opt ; expression-2opt ; expression-3opt ) statement
switch ( expression ) statement
case expression : statement
default : statement
break ;
continue ;
return ;
return expression ;
goto identifier ;
identifier : statement
;
```

18.4 External definitions

```
program:
external-definition
external-definition program

external-definition:
function-definition
data-definition

function-definition:
type-specifieropt function-declarator function-body

function-declarator:
declarator ( parameter-listopt )

parameter-list:
identifier
identifier , parameter-list

function-body:
type-decl-list function-statement

function-statement:
{ declaration-listopt statement-list }
```

~1978

Multics (PL/I)

↓

Unix (C)

↓

Linux (git)