



**iOS Theory Pdf**  
**- Dhwani**

IT - 8

DECEMBER  
2019

SATURDAY 21

(355-010) Week 51

optionals.	18 Dec
protocol, extension	17 Dec, 2 Feb
ios Arch.	13 Dec
ios App Life Cycle	07 Dec
ios MVC	04 Dec
Set, Array, Dictionary	03 Dec
SQLite func.	30 Nov
App to appstore	29 Nov
picker view	27 Nov
JSON parsing	23 Nov
Core Data vs SQLite	20 Nov
view controller	19 Nov, 6 May
user default	17 April
inspector type	19 Feb
sqlite	23 Feb
IBOutlet, IBAction	08 Feb
table view	03 June
core data.	18 Nov
Geo - reverse Coding	15 Nov. SUNDAY 22
Storyboard	13 Nov.
Collection view	12 Nov.
Interface Builder	08 Nov.
Auto Layout	07 Nov.
Mapkit view Framework	04 Nov.
Annotation	02 Nov
Callouts	01 Nov

JANUARY 2020						
S	M	T	W	T	F	S
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

18 WEDNESDAY

(352-013) Week 51

DECEMBER  
2019

## Optionals in Swift.

Data type as Optional by appending  
! / ? to the type.

If an optional contains a value in it,  
it returns value as Optional <Value>

If it not → it returns NIL.

- 2 types. Declare.

Implicit → !

Explicit → ?

var v1 : String!

v1 = "Dheerani"

print(v1!)

var v2 : String?

v2 = "Welcome"

print(v2!)

→ type that represent value that may  
either be present or absent.

DECEMBER 2019

M	T	W	T	F	S	S
30	31				1	
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29

ex.

protocol instrument {

var name: String & get set<sup>3</sup>

var numOfString: Int & get<sup>3</sup>

func play()

DECEMBER  
2019

3

TUESDAY

17

(351-014) Week 51

## Protocol & Extension.

Protocol let us define contract that conforming types must adhere to.

- defines "blueprint" of methods or properties that can be adopted by classes, structure

- use to define set of rules:  
[See on 2nd Feb 2019]

Extension → allows you to add new functionality to an existing type,

including types defined in Swift's standard library without modifying the original source code.

ex.

extension Int {

var square: Int {

return self \* self

3

3

let num = 5

print(num.square) → O/p : 25

JANUARY 2020

S	M	T	W	T	F	S
				1	2	3
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

~~method / property on blueprint. & classes can adopt it.~~

FEBRUARY 2019 ~~collection of some special task / functionality~~ SATURDAY 02  
✓ 01 2019 21/21 (tailored) discussion  
with 20. (033-332) Week 05

// Protocol. → define a blueprint of methods / properties that can be adopted by classes

↳ collection of methods, properties & other specifications that are tailored to a special task or piece of functionality.

code: protocol prot 2  
↳ func show() ↳

protocol prot 2  
↳ func disp() ↳

class abc : prot1, prot2 &  
func show()  
↳ print ("show method of prot1...") ↳ SUNDAY 03

func disp()  
↳ print ("disp method of prot2...") ↳ ↳

var obj = abc()  
obj.show()  
obj.disp()

S	M	T	W	T	F	S
31	1	2				
1	2	3	4	5	6	7
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30						

MARCH

APRIL

$$a \sin \theta = c \sin \theta$$

08

FRIDAY

(039-326) Week 06

IBOutlet: ✓

FEBRUARY  
2019

right click on view controller open  
seg.

right click on screen label →  
new referencing outlet dragging to  
view controller class then click

give label name.

→ C <sup>outlet</sup> Interface builder is helpful stands for.

C ! means → no need to assign any  
thing

In view DidLoad method ()

↓

lb.desct = "iOS..."

then run.

⇒ use to create reference to user interface  
element (e.g. button, label, textfield)  
in view controller class.

FEBRUARY 2019

M	T	W	T	F	S	S
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	01	02	

→ this connection allows view  
controller to access & manipulate  
properties of user interface

element programmatically.

IBOutlet

FEBRUARY  
2019

IBAction : ✓

SATURDAY 09

(040-325) Week 06

Button right click.

touch up Inside drag to after it.  
name change.

select it. text cut & paste here.  
run.

⇒ stands for Interface Builder Action.

→ used to create a method / function in view controller class that responds to user interaction with user interface element. (eg. button tap)

→ connection allows view controller to handle event triggered by user interface element such as SUNDAY 10  
button taps, text field edits.

MARCH

MARCH 2019

S	M	T	W	T	F	S
31					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

APRIL

IBAction

Q. Notes on Inspector types -

Q

FEBRUARY  
2019

shs

TUESDAY

19

(050-315) Week 08

utility Area Inspector → (F) type.

file : where your actual file is located.

history : navigate file

quick help : in editor whatever component we select → quick info. get.

Identity : show connection between screen & class.

↳ open screen add select object

class object ↳.

↳ chain behavior of source code  
Gives each ui of this class  
screen object.

↳ run time (for update seg.)

→ this class file seg. ↳.

↳ Segie → multiple connection

→ seg. / name

segie.

→ screen → connect seg.

MARCH 2019

S	M	T	W	T	F	S
31	1	2				
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

MARCH

APRIL

20 WEDNESDAY

FEBRUARY  
2019

(051-314) Week 08

to pass data or textbox to specific screen magic is used.

→ in view controller ...

after class .. func

waite prepare ender & logic

Size Inspector : Specific ui control  
size control



Connection : Show design type / time connection.



Key Value

Attribute :

FEBRUARY 2019

M	T	W	T	F	S	S
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28			

Q2

## SQLite.

FEBRUARY  
2019

DataBase...

SATURDAY

23

(054-311) Week 08

- ↳ DB that can be used by apps in iOS (also used by iOS) is called SQLite.
- it's relational database.
- It's stored in C-library that's built into the software you're about to use.
- There is no stand-alone SQLite server operating in background like SQL Server, Oracle, MySQL.

## Features:

- open source
- fast
- fully supported by Apple,

SUNDAY 24

as it's used in both iOS & Mac OS.

- contains an embedded SQL engine, so almost all of your SQL knowledge can be applied.

MARCH 2019

S	M	T	W	T	F	S
31					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

- works as part of app itself so doesn't require extra active service.

MARCH

APRIL

Q

User Defaults.

17

WEDNESDAY

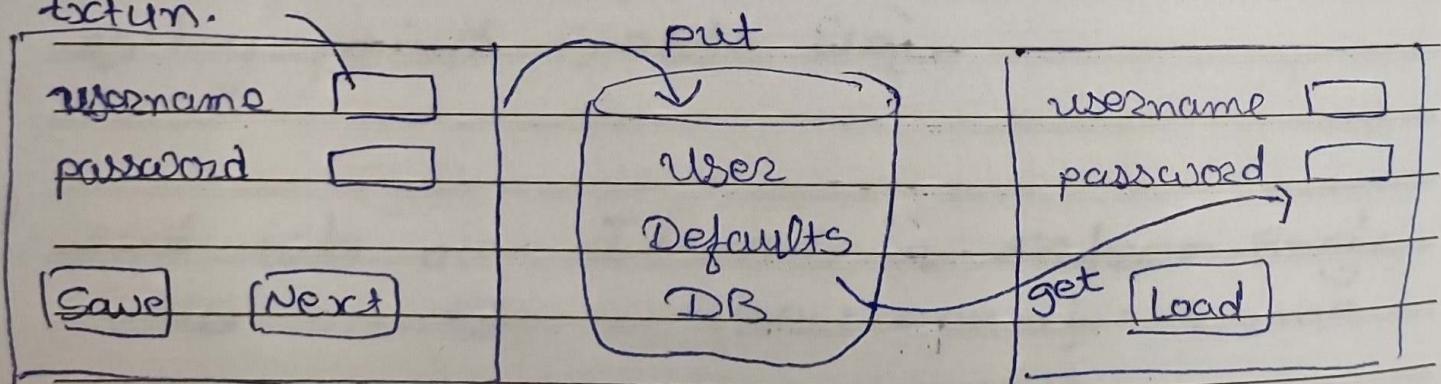
APRIL  
2019

(107-258) Week 16

ex. login, reg

- used to store small pieces of data which persist across app launches.
- very common to use to store app settings / user preferences.
- let you store key value pairs, where key is always String and value can be one of the following data types:  
String, Num, Date, Array, Dictionary

txtun.



Main Screen

Second Screen

Q. Which datatype we can store in user default?

APRIL 2019

→ String, Num, Date, Array,

Dictionary

M	T	W	T	F	S	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	1	2	3	4

sidet summary

MAY  
2019

MONDAY 06

(126-239) Week 19

View Controller: instance of 'UIViewController' class or its subclass.

→ typically controls a single screen / view

\* => Short Step for share data

1. Create 2 view controller in storyboard, one for each screen

2. Create property in 1st vc to hold the data that needs to be shared

3. In 1st vc → create segue to 2nd vc using Interface Builder.

4. Implement prepare (for: sender;) method in 1st vc to pass data to 2nd vc before segue is performed

5. In 2nd vc → create property to hold data received from 1st vc & update UI accordingly.

JUNE

JULY

AUGUST

JUNE 2019

S	M	T	W	T	F	S
30	1	2	3	4	5	6
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29

# View Controller

19

TUESDAY

NOVEMBER  
2019

(323-042) Week 47

- used to manage your UIKit app's interface.
- every app has atleast one view controller. but mostly have many more.
- A view controller manages a single root view, which may itself contain any num of subviews.
- User interactions with that view hierarchy are handled by your view controller which coordinates with your other objects of app as needed.

Steps to Share Data betw 2 Screen

- 1) create 2 view controller in storyboard one for each screen.
- 2) create a property (prepare function) in 1st view controller to hold the data that need to be shared.

NOVEMBER 2019

M	T	W	T	F	S	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

See 6 May.

13

FRIDAY

iOS Architecture.

DECEMBER  
2019

(347-018) Week 50

### Cocoa Touch (APP)

UIKit

Mapkit

Gamekit

...

### Media

AVFoundation

Core Graphics

OpenGL

Core Image

### Core Services

Core Data

Foundation

Quick Look

Core Location

Address Book

...

### Core OS

Accelerate

Security

OpenCL

System Configuration

...

DECEMBER 2019

M	T	W	T	F	S	S
30	31			1		
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29

DECEMBER  
2019

THURSDAY

12

(346-019) Week 50

## Cocoa Touch Layer

- contains key framework for building iOS apps.
- this framework define appearance of your app.
- Drives the UI.  
provides controller, widgets, ...
- Provide access to main function system  
Contact, Camera, touch input,  
push notification, etc
- Should be the first place you look  
for any features you want to  
use in your app.
- include high level features. &  
cocoa touch framework.

feature :

AirDrop , Auto Layout  
Storyboard , Gesture Recognizer  
Multi tasking , Standard System-  
view controller

JANUARY 2020

S	M	T	W	T	F	S
				1	2	3 4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

11

WEDNESDAY

(345-020) Week 50

DECEMBER  
2019

framework features:

Gamekit , UIkit } Framework  
Map kit , Message UI  
Event kit UI , Addresskit UI

Media Layer.

- the graphics libraries live here:  
Core Graphics      Core Text  
Open GL      Image I/O  
Photo Library      Assets Library  
Animation

- Audio

Media player      Core Audio  
Open AL

- AirPlay

DECEMBER 2019

M	T	W	T	F	S	S
30	31	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29				

DECEMBER  
2019

TUESDAY 10  
(344-021) Week 50

## Core Layer Services

- give access to fundamental resources needed for app.
- Built on Core Foundation + framework.

### Features:

- iCloud Storage
- File Sharing Support
- SQLite
- XML Support
- Peer-to-Peer Service
- Core Location
- Accounts
- Core Data
- Core Media
- Address Book

## Core OS Layer

- probably not going to use these in your app until & unless we make advance version app then Core OS is used which provide us Bluetooth, USB, Security, other accessories.

JANUARY 2020

S	M	T	W	T	F	S
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

# 09

MONDAY

(343-022) Week 50

DECEMBER  
2019

features.



home Screen

framework



Included App

Accelerate

Multitasking

Core Bluetooth

Siri

Generic Security

Game Center

64 bit Support

Jail Breaking

browsing

SUNDAY

DECEMBER 2019

M	T	W	T	F	S	S
30	31			1		
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29

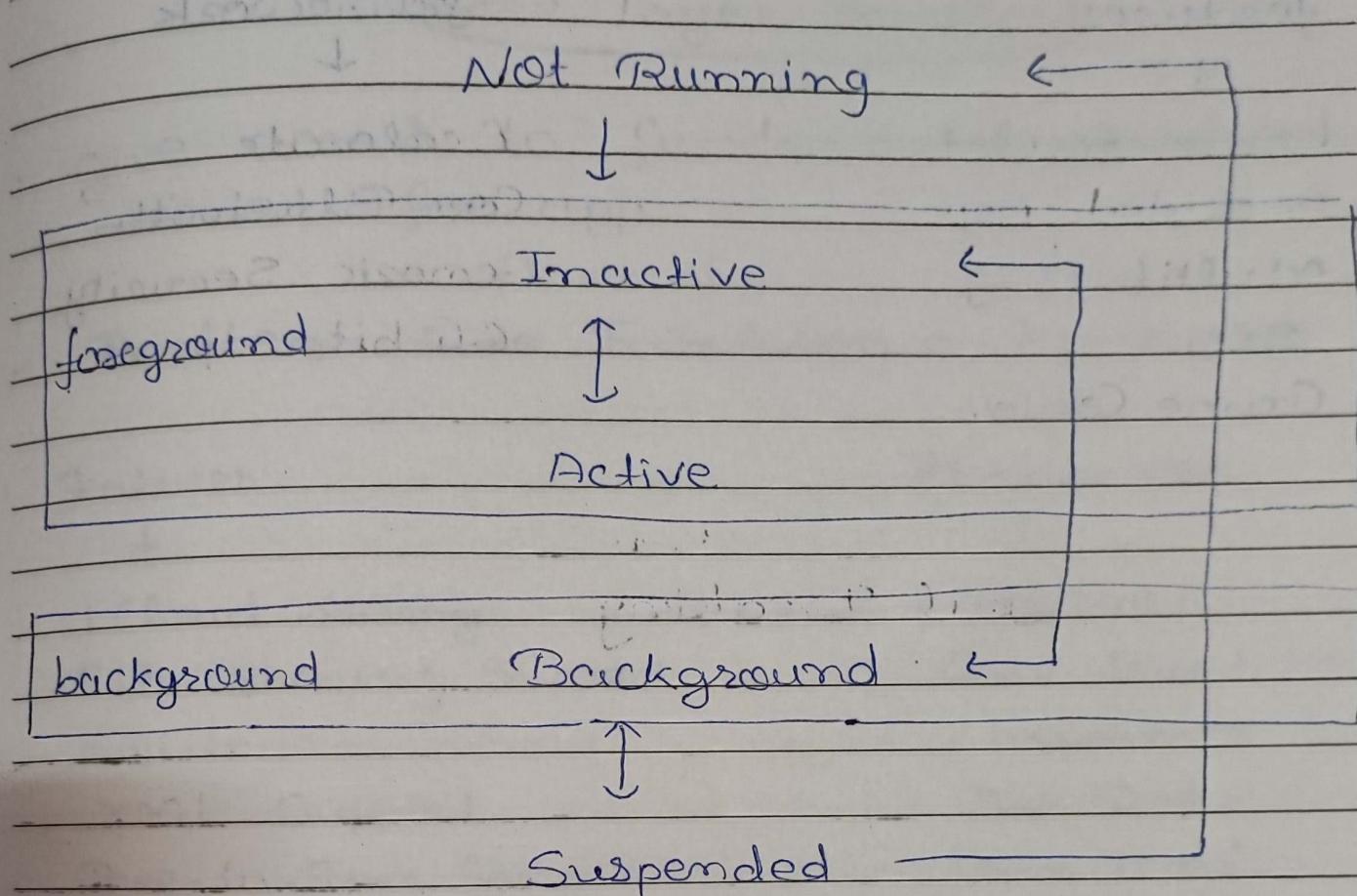
DECEMBER  
2019

# iOS App. Life Cycle.

SATURDAY

07

(341-024) Week 49



## ① Not Running

SUNDAY 08

- App is not yet started / is terminated by system.

## ② Inactive

- App is active but it is not being used → called inactive.

JANUARY 2020

S	M	T	W	T	F	S
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

# 06

FRIDAY

(340-025) Week 49

DECEMBER  
2019

→ it is short transition state where app is actually in the foreground but not receiving events.

→ this happens when a message / call received.

### ③ Active imp state

→ App is in the active state running in foreground & receiving events.

### ④ Background

→ When user presses Home button / switches to another app, the app transitions to this state.

→ It can continue to execute code & perform tasks in background,

DECEMBER 2019

M	T	W	T	F	S	S
30	31				1	
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29

such as updating the user interface, downloading data, playing audio.

DECEMBER  
2019

05  
THURSDAY

(339-026) Week 49

(5) Suspended

→ if the app does not request additional background execution time or perform specific background tasks, it is suspended by the system.

→ In this state, the app remains in memory but doesn't execute any code.

→ It helps to protect system resources & battery life.

JANUARY 2020

S	M	T	W	T	F	S
				1	2	3
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

Blue → use / represent

Black → Interact.

# 04

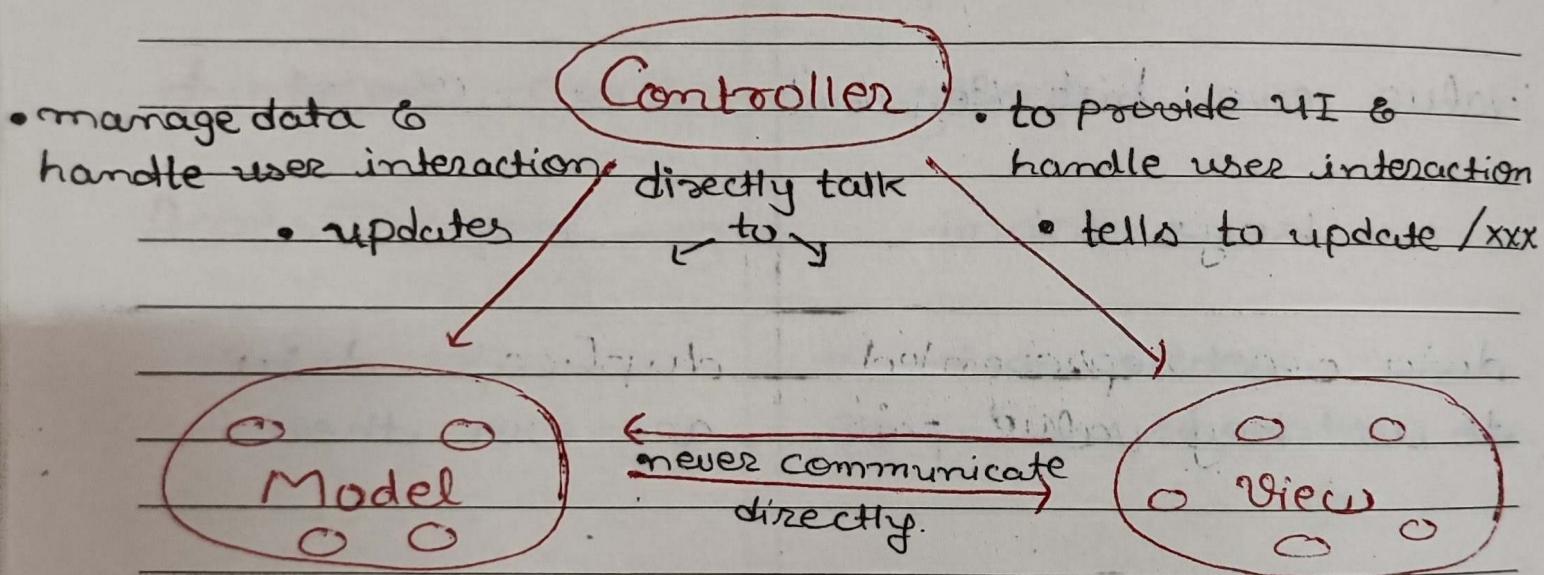
WEDNESDAY

MVC

DECEMBER  
2019

(338-027) Week 49

- responsible for interpreting user input, updating the model accordingly, updating view to reflect any changes in model.
- use to update our Model & Trigger our View
- intermediary between Model & View



- represent "state, data, info." in app.
- wrapper of data
- responsible for
- representation of UI

DECEMBER 2019

M	T	W	T	F	S	S
30	31			1		
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29

managing app's data & ensuring that it remains consistent & up to date.

- responsible for displaying app's data to user & capturing user's input



# 30 Core SQLite Functions.

SATURDAY

NOV/DEC  
2019

(334-031) Week 48

sqlite 3 - open() open db connection  
close() close  
prepare - v2() prepare SQL stmt.  
step() step through result  
finalize() finalize stmt  
exec() execute SQL statement  
column-<type>()

01 SUNDAY

NOVEMBER 2019

M	T	W	T	F	S	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

# 5 steps to publish App to App Store.

NOVEMBER  
2019

FRIDAY

29

(333-032) Week 48

## \* Precondition

1. project that is ready for submission & passes Apple's app store guidance.
2. valid apple developer account.
3. Computer that runs Mac OS X.
4. the following programs installed:

Xcode, Keychain Access.

## 7. Make the Release Build

Package actual binary that users will be uploading to the store.

## 8. Fill in Version Information

Users will decide whether or not to install your app based on your store listing.

## 9. Submit Version for Review

Must reviewed by Apple team before release any app.

## 10. Release

DECEMBER 2019

S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

DECEMBER

28

THURSDAY

(332-033) Week 48

NOVEMBER  
2019

Step.

1. Assemble App Store Info.

u. imp to collect all info you need to submit before you start process.

2. Create a Bundle Identifier

it is the name of your app, as seen by both App Store & iOS devices

3. Create a Certificate Signing Request

used to link your computer to your Apple developer account

4. Create App Store Production

Certificate

Code signing certificates are used to link iOS apps to your Apple developer account

5. Create a Production Provisioning Profile

are packaged with iOS apps so user devices can install them,

NOVEMBER 2019

M	T	W	T	F	S	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

6. Create an App Store Listing

reserve a slot in App store for your app for users to see.

# Picker View.

NOVEMBER  
2019

WEDNESDAY 27

(331-034) Week 48

- It is an instance of UIPickerView class which inherits UIView.
- used to make a selection from multiple choices.
- Picker View is a user interface element in iOS that allows user to select a value from a list of predefined options.
- It consists of one or more columns, each containing a set of rows with values that the user can select by scrolling.

## \* Implementation \*

- 1) Create new Xcode project & select "Single View App" as template.
- 2) Open Main.Storyboard file & drag a UIPickerView onto the view controller.

DECEMBER 2019

S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	31	31	31	31

DECEMBER

26 TUESDAY

(330-035) Week 48

NOVEMBER  
2019

3) Create an IB Outlet for UIPickerView view in view controller file.

4) Set the data source & delegate of UIPickerView view to the view controller in viewDidLoad() method.

pickerView.dataSource = self  
delegate

5) Implement UIPickerViewDelegate & UIPickerViewDataSource protocol in view controller file.

& write code then execute.

24 SUNDAY

NOVEMBER 2019

M	T	W	T	F	S	S
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	

# JSON + Parsing + ex.

23

SATURDAY

(327-038) Week 47

NOVEMBER  
2019

JSON → JavaScript Object Notation  
(programming language)

- It is more popular today due to its simplicity, lightweight nature & widespread support across different prog. lang. & platform.
- Lightweight format for storing and transporting data.
- JSON is often used when data is sent from a server to a webpage.
- Alternative to XML
- provides support to parse JSON object & array.

24 SUNDAY

\* Advantage of JSON over XML

- 1) JSON is faster, safer, easier, flexible, than XML.

NOVEMBER 2019

M	T	W	T	F	S	S
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	26	27

- 2) Unlike XML, it's shorter & quicker to read & write.

Completed

Completed 6

NOVEMBER  
2019

22 + Friday + 1025

FRIDAY 22

(326-039) Week 47

- 3) json is lightweight & simplicity than XML so it has smaller msg size & runs faster, consume less memory.
- 4) it uses curly to store data while XML stores data in trees using curly brackets in JSON.

XML → Extensible Markup Language

- text based markup language used to store data while transport it.
- similar to HTML C used to display data, XML used to describe data in application.

high level way of handling data

Parsing JSON

SUNDAY

mix your code: JS, javascript

S

{"name": "John", "age": 22, "city": "New York"}  
"age": 22, INT value.  
"city": "New York", STRING

3. JSON is a subset of JavaScript object notation.

key (String)                  values  
                                  (JSON type)

DECEMBER 2019

S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

DECEMBER

21 THURSDAY

NOVEMBER  
2019

(325-040) Week 47

↳ JSON obj contain key / value pairs like map.

↳ { } (curly brace) represent json obj.

↳ [ ] (square bracket) represent json array.

[{"Sunday": "Monday", "Monday": "Tuesday"}] object map

↳ [Sunday, Monday, Tuesday] list items

↳ Parsing XML

```
<person>: <name> (is Dheeraj) </name>
    <name> Dhani Parmar </name>
    <age> 22 </age>
    <sex> male </sex>
</person>
```

NOVEMBER 2019

M	T	W	T	F	S	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

# Core Data vs SQLite

NOVEMBER  
2019

WEDNESDAY 20

(324-041) Week 47

## SQLite

1) have data constraints feature.

2) operates on data stored on disk.

3) can drop table to edit data without loading them in memory.

4) slow as compared to Core Data.

## Core Data

1) don't have. (need to implement by business logic)

2) operates in memory. (data needs to load from disk to memory)

3) need to load data (entire) if we need to drop or update table.

4) fast in term of record creation.

DECEMBER 2019

S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

DECEMBER

# Core Data Stack

NOVEMBER  
2019

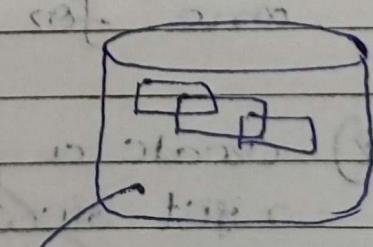
MONDAY 18

(322-043) Week 47

- framework provided by Apple to save, track, filter, modify data within iOS app.
- it is not database; it uses SQLite as its persistent store.
- need to manage model layer object in our app.
- manages object graph; track the changes in database & modifies data on user interaction.
- framework that managing the lifecycle of object in your app.
- based on MVC pattern.
- introduce few new basic concepts & terminology.

APP

Context



Persistent  
Container

DECEMBER 2019

S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	8S	15S	22S	29S

DECEMBER

- contains low level info (name, location, type of datastore)
- app will have 1 instance of this for each db that it needs to interact with.

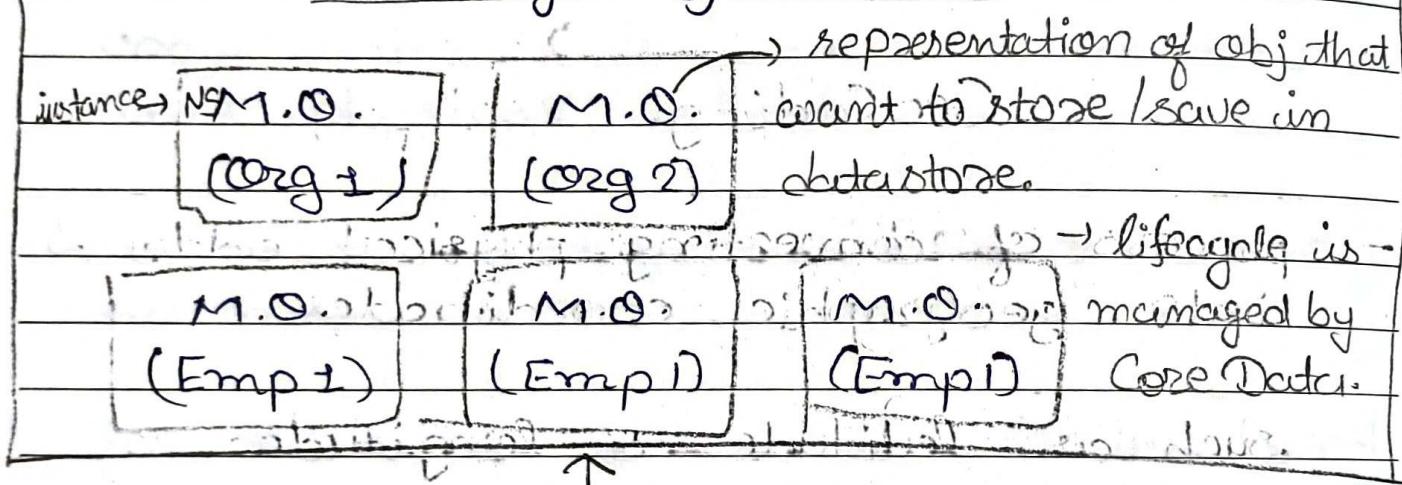
**16** SATURDAY

NOVEMBER  
2019

(320-045) Week 46

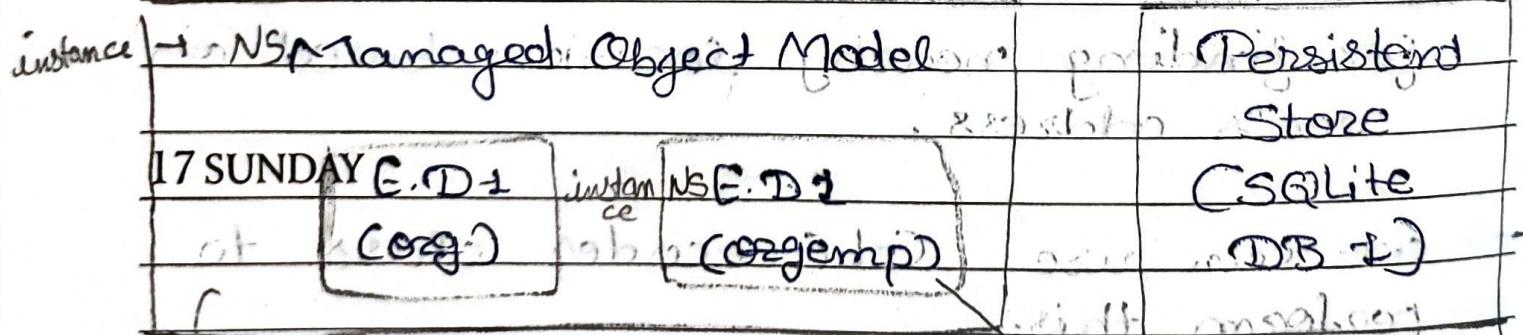
- buffer bet<sup>n</sup> app & datastore
- inside this context you can: add, delete, modify managed obj. → to perform CRUD I need to call methods here!

instance → NS Managed Object Context



instance → NSPersistentStoreCoordinator

- represent connection to datastore
- used by M.O.C



- collection of entity description
- a file that ends with .xcdatamodeld
- modelId. extension
- proj compiled → file
- doesn't contain data actual

NOVEMBER 2019

M	T	W	T	F	S	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

→ contains low level info (name, location, type of datasource)

→ app will have 1 instance of this for each db that it needs to interact with.

16 SATURDAY

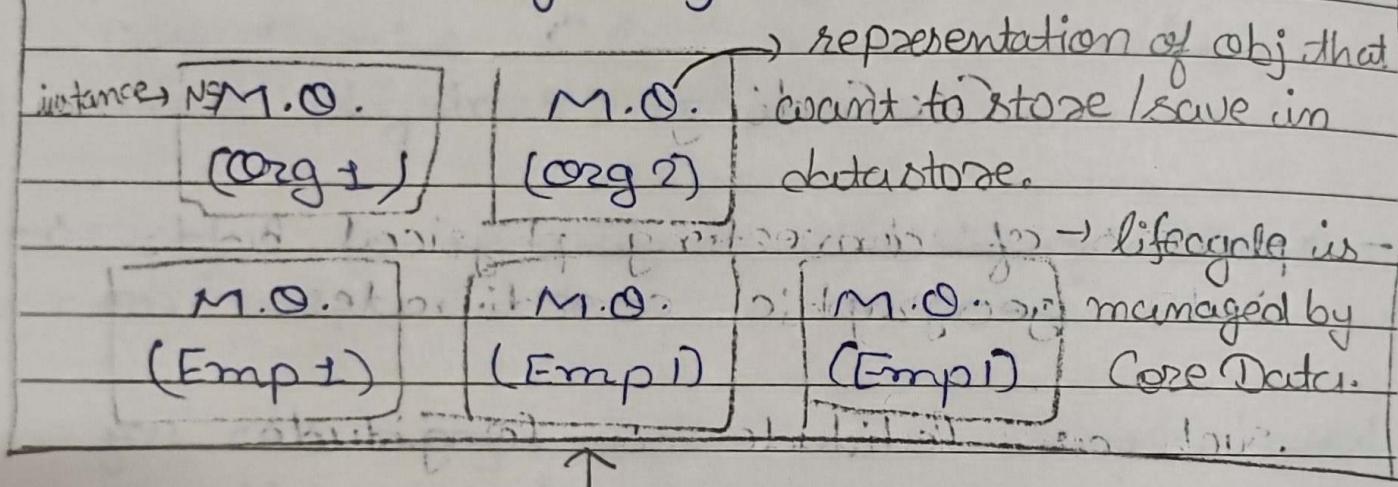
NOVEMBER  
2019

(320-045) Week 46

→ buffer bet<sup>n</sup> app & datastore.

→ inside this context you can: create, delete, modify managed obj. → to perform CRUD I need to call methods here.

instance → NS Managed Object Context



instance → NS Persistent Store Coordinator

→ represent connection to data store  
→ used by M.O.C

instance → NSManagedObject Model

17 SUNDAY E.D.  
(Org)

instance → NSE.D.  
(OrgEmp)

Persistent Store  
(SQLite DB +)

→ collection of entity description

→ a file that ends with .xcdatamodeld

NOVEMBER 2019 modelId. extension

M	T	W	T	F	S	S
1	2	3	4	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	

describe table within db

→ db tables called entities

→ similar to schema for db

→ doesn't contain data actual

→ proj compiled to file

compiled to mom file

# Geo Coding.

NOVEMBER  
2019

Reverse Geo Coding.

FRIDAY

15

(319-046) Week 46

- features of iOS Mapkit framework that allows you to convert between geographic coordinates & physical addresses & vice versa.

Geocoding API

## Geo Coding

- process of converting physical address into geographic coordinates such as latitude & longitude.
- useful for displaying location of an address on a map or for finding nearby location based on user's address.
- we can use GLGeocoder class to perform this.
- simply provide address info to GLGeocoder obj & it returns corresponding geographic coordinates in form of CLLocation obj.

DECEMBER 2019

S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

DECEMBER

14 THURSDAY

(318-047) Week 46

NOVEMBER  
2019

## Reverse Geocoding

- opposite process of Geocoding.
- involves converting geographic coordinates (longitude & latitude) into physical address.
- useful for displaying address of location on map or for obtaining info about location based on its coordinates.
- process of converting CLocation into CLPlacemark.

NOVEMBER 2019

M	T	W	T	F	S	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

NOVEMBER  
2019

## Storyboard.

WEDNESDAY 13

(317-048) Week 46

- visual representation of user interface of an iOS app consisting of a sequence of screen / view controller interactions that user navigates through to accomplish a task or access content.

→ Storyboard are created & edited using Interface Builder, a graphical tool in Xcode.

### Advantages

- visual representation
- faster development
- collaboration
- Segues

### Disadvantages

- large files
- merge conflicts
- limited reuse
- storyboard versioning

DECEMBER 2019

S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

DECEMBER

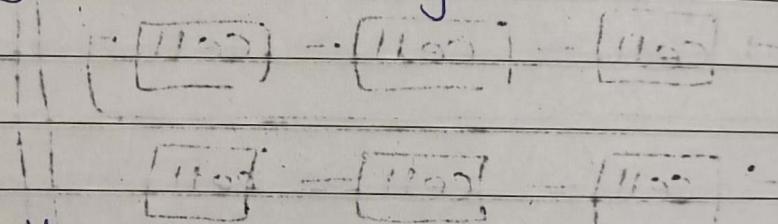
# 12

TUESDAY

## Collection View.

(316-049) Week 46

NOVEMBER  
2019

- used to customise layout in any way we wish.
  - its like Grid view.
  - similar to UITableView except the fact that it's more customizable.
  - but difference is UICollectionView can display more than 1 column & supports horizontal scrolling.
  - Components : 
- UICollectionViewCell
- just like UITableViewCell, cell are subviews of UICollectionView
  - our content is displayed in this cell only.
  - cell are dequeued as user leaves screen.

NOVEMBER 2019

M	T	W	T	F	S	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

NOVEMBER  
2019

MONDAY 11

(315-050) Week 46

## Supplementary views

- consist of other imp component such as labels, section, header, footer
- that are used to define & divide content area.

header + [cell]

scrolling direction

→ [cell] - [cell] - [cell]

[cell] - [cell] - [cell]

footer + ]

steps : + ex.

## DECEMBER 2019

S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

DECEMBER

NOVEMBER  
2019

## Interface Builder.

FRIDAY

08

(312-053) Week 45

- graphical tool in Xcode.
- IDE [Integrated Development Env.] for iOS & macOS app dev.
- enables developers to design & lay out the UI [User interface] of an app visually, without having to write code.
- Interface Builder provides drag-&-drop interface for adding UI elements such as button, text field, images to a storyboard.
- Also enable developers to create connection between UIElements & code using IBAction & IBOutlet.
- By control dragging from UI component element to code file,

DECEMBER 2019

S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	28	29	30	31

DECEMBER

# 07

THURSDAY

## Auto Layout.

(311-054) Week 45

NOVEMBER  
2019

developer can create these connection & write code that respond to user input / update UI dynamically.

+  
=

- supports auto layout.

[A powerful system for laying out UI elements that enables them to adapt to different screen size & orientation.

### Constraint based Layout.

- with AutoLayout, developer can create a UI that looks great on any device from smallest iPhone to largest iPad.

### Why AutoLayout?

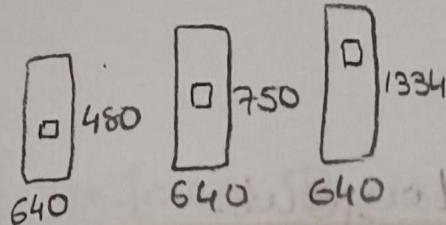
- without using auto layout, position of button,

we layout in storyboard is fixed.

NOVEMBER 2019

M	T	W	T	F	S	S
1	2	3	4	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	

PSC.

NOVEMBER  
2019

WEDNESDAY

06

(310-055) Week 45

- In other words,  
we hard code the frame origin  
of button.
- ~~ex~~ "Hello World" button frame  
origin is set to (120, 270)
- therefore, whether you are  
using 3.5 inch or  
4.7 inch simulator,  
iOS draws the button in  
specified position only.
- => Auto Layout is all about Constraint.
- Constraint in Auto layout are  
expressed in Mathematical form.

DECEMBER 2019

S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	31	31	31	31

# 05

TUESDAY

(309-056) Week 45

NOVEMBER  
2019

- XCode provide 2 type of way to define auto layout constraint.

- ① Auto layout bar (assistant tab)
- ② Control-dragging (ctrl + click)

auto layout - fix spacing after  
constraint tab - space 60px  
1. 100px 2. 20px 3. 30px  
3 additional points

align height, most align at  
constraint

and your own width & height of view  
design based condition, mitigation  
and mitigation condition or

NOVEMBER 2019

M	T	W	T	F	S	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

NOVEMBER  
2019

# MAP KIT VIEW framework

MONDAY

04

(308-057) Week 45

Purpose:

- to provide developers with tools to integrate interactive maps into their app easily.
- with mapkit, developers can easily maps, add annotations like pin pins & callouts, show location & provide features like route planning & searching for place.
- In simple term Mapkit helps developers create apps that use map for navigation, location based service & visualization geographic data.

DECEMBER 2019

S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

DECEMBER

02

SATURDAY

(306-059) Week 44

NOVEMBER  
2019

## Annotation

- in Mapkit are object used to represent points of interest on maps, such as landmark, building, user-defined location, businesses.
- An annotation is represented by instance of MKAnnotation protocol which contains latitude & longitude of the location as well as title & subtitle that describe annotation.  
- also include custom image & callout.

03 SUNDAY

NOVEMBER 2019

M	T	W	T	F	S	S
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	

01

FRIDAY

(305-060) Week 44

NOVEMBER  
2019

## Callouts.

- Callouts is a graphical object that is displayed when a user taps on an annotation.
  - provides additional info. about location that annotation represent.
  - By default, callout displays title & subtitle. of annotation also include but it can<sup>↑</sup> custom view & button.
  - represented by an instance of MKAnnotationView class, which is subclass of UIView.
- ⇒ Together, Annotation & Callouts provide a way to display & interact with info. on map in iOS.

NOVEMBER 2019

M	T	W	T	F	S	S
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	

- A : mark specific location
- C : provide additional detail about each location.