Print (“Something else”)

}

Structs are templates - contains two fundamental things

**Attributes** hair color, height, and birthday (let attributes )

**Actions** change hair, brush teeth (var actions)

Varibel in App

File>new>project>IOS>App>give the product a name First app, team none, organization identity (domain name put anything for now JacksonGuentherSwift1), interface - swift, testing to none and storage

(Do not click source control box)

Xcode Variables - Var can be changed in the future and Iet cannot

Methods - string

Logic - print

Single = statement, double == question != not equal to

Change “” to Inch marks on xcode

Import UIKit

Var greeting = “hello, playground”

Greeting = “hole”

Iet birthMonth = 12

Var congrats = “I am happy for you, your birth month is ” +String( birthMonth)

Iet cuurentMonth = 1

Iet countdown + birthMonth - currentMonth

If countodwn < 5 {

print(“your birthday is coming up!”)

} else }

a strut is called an attribute

**2/4 - in class coding**

Import UIKit

Struct Contentview: View {

@state var numberofClicks = 900

Var body: some View {

Var greeting = “hello, playground”

Struct person { let birthMonth: Int

var firstName: String

var numberOfLungs = 2

Func sayHello() { print(“good morning” + self.firstName) print(“good to see you”)

}

MUTATING Func removeLungs() {

self.numberofLungs = self.numberoflungs - 1

Print(self.Firstname + “ “

}

Var Artie = Person(birthMonth: 12, firstname: “Artie”)

Var ray = Person(birthMonth 6, firstName: “Raymond”, numberOf Lungs:1)

Artie.firstname = “Kevin”

print(artie.birthmonth) artie.firstName

Artie.sayHello()

Ray.sayHello()

Ray.removeLung(1)

Practice

Struct contentView {

Var body: some View {

Text(“hello there!”)

.fontweight(.bold)

@state var numberofclicks = 90

Button(action: () -> void

**2/6 - in class coding - struct/list**

Import UIKit

Struct GroceryItem {

Var groceryItem:string

Var groceryEmoji:string

Var groceryAisle:string

Struct Contentview: View {

@state var grocerylist = [GroceryItem(groceryName: “Bacon”, groceryEmoji:” “ ,

groceryAisle: “Back fridge area”), Grocery Item: “Eggs” , groceryEmoji: “ “ , groceryItem: “Dairy”),

GroceryItem(groceryName: “eggs” , groceryEmoji: “ “ ,

groceryAisle: “5B” )

]

Var body: some View {

List(0..<groceryLists.count, id: \.self) { Index in Vstack

Var body: some View {

List(0..<grocrylist.count, id: \.self { index in

text(grocryList[index])

Lists {

Text(“bacon”)

Text(“eggs”)

text(“cofee”)

text(“milk”)

}

}

}

#Preview {

ContentView()

}

**2/11 - 2nd page of code from 2/9**

Import UI

Struct detailedView: View

@state var selectedindex = 1

Var body: some View {

text(“ “)

navigationView {

} like an a tag (big)

NavigatioLink(destination:detailview(selectedIndex: Index))

**First page**

By making this code global ( you can use selectindex to pass on info

@state var grocerylist = [GroceryItem(groceryName: “Bacon”, groceryEmoji:” “ ,

groceryAisle: “Back fridge area”), Grocery Item: “Eggs” , groceryEmoji: “ “ , groceryItem: “Dairy”),

GroceryItem(groceryName: “eggs” , groceryEmoji: “ “ ,

groceryAisle: “5B” )

]

**2/13 -**

**(first page code) how to design your apps logo on xcode**

struct ContentView:

View var body: some View

{ NavigationView

{ List(0..<groceryList. count, id: \.self) { index in NavigationLink(dest athcs:detailView（selectedIndex: index））

｛ VStack

{ HStack

{ Text (groceryList[index] groceryEmoji)

if groceryList[index].inCart == false {

Text (groceryList[index]-groceryName)

• fontWeight(• semibold)

} else {

Text (groceryList[index]-groceryName)

• foregroundColor (Color.gray)

Text (groceryList[index]-groceryAisle)

• fontWeight(. light)

• foregroundColor

}

}

}

**Second page code**

Strut detailedView: View {

@state var selectIndex = 1

Var body: some View {

VStack {

Text(grocoeryList

[slectedIndex].groceryName)

.font(.LargeTitle)

.fontweight(.black)

Text(groceryList

[selectedIndex]. groceryAisle)

.fontweight (.ultraLight)

.foregorundColor(Color

.white)

.shadow

**2/18 - circle**

**2/20 -** make a file on your phone. Use a database in your app. User defaults (built in)

UserDefaults.standard.integer(forKey: “name”)

I send a green text message but delete

**2/25**

struet GroceryItem: Codable {

Func saveGroceryItem(gorceryListToBeSaved : [groceryItem]) {

If let encoded = try? JSONEncoder().encode(groceryListToBeSaved) {

UserDefaults.standard .set(Encoded, Forkey: “savedGroceryList”)

}

}

**Detailed view below**

import SwiftUI

struct detail view: View {

@State var selectedIndex = 1

Binding var groceryList:

Var body: some View {8

VStack(alignment: • leading) {

ZStack {

Image(groceryList[selectedIndex].groceryImage)

VStack {

Text(groceryList[selectedIndex].groceryName)

• font(. largeTitle)

• fontWeight(black)

• foregroundColor (Color white)

•shadow(color: •black, radius: 5, x: 0, y: 5)

Button (action: {

groceryList[selectedIndex].inCart = false {

groceryListIselectedIndex]. inCart = true

}

else {

groceryList[selectedIndex].inCart = false

Label: 1

if groceryList[selectedIndex]. inCart = false {

Text("Put in Cart")

｝

else {

Text ("Remove from Cart")

**Other view below**

struct

GroceryItem

}

• First App 1 0 iPhone 16 Pro

Si detailView.swift

Build Succeeded | 2/18/25 at 11:59 AM

struct ContentView: View {

@State var groceryList = [

GroceryItem(groceryName: "Bacon" fridge area", groceryImage:

groceryEmoji: "bacon" ),

groceryAisle: "Back

GroceryItem(groceryName: "Eggs", groceryEmoji:

"Dairy", groceryImage: "eggs", inCart: true),

groceryAisle:

GroceryItem(groceryName: "Coffee",

]

"5b", grocery Image: "coffee")

groceryEmoji: "®", groceryAisle:

var body: some View {

NavigationView {

List(0..«groceryList.count, id: I.self) & index in

NavigationLink(destination:detailView(selectedIndex: index,

groceryList: $)) {

VStack {

HStack {

Text (groceryList[index].groceryEmoji)

if

groceryList[index]. inCart = false {

Text (groceryList[index].groceryName)

**3/11 - Attractions is used for “stuff inside stuff “/tab bars (bookmarks apple)**

Struct Attraction {

Var name: string

Var type: string

}

StructCity {

Var name: String

Var attraction: [Attraction]

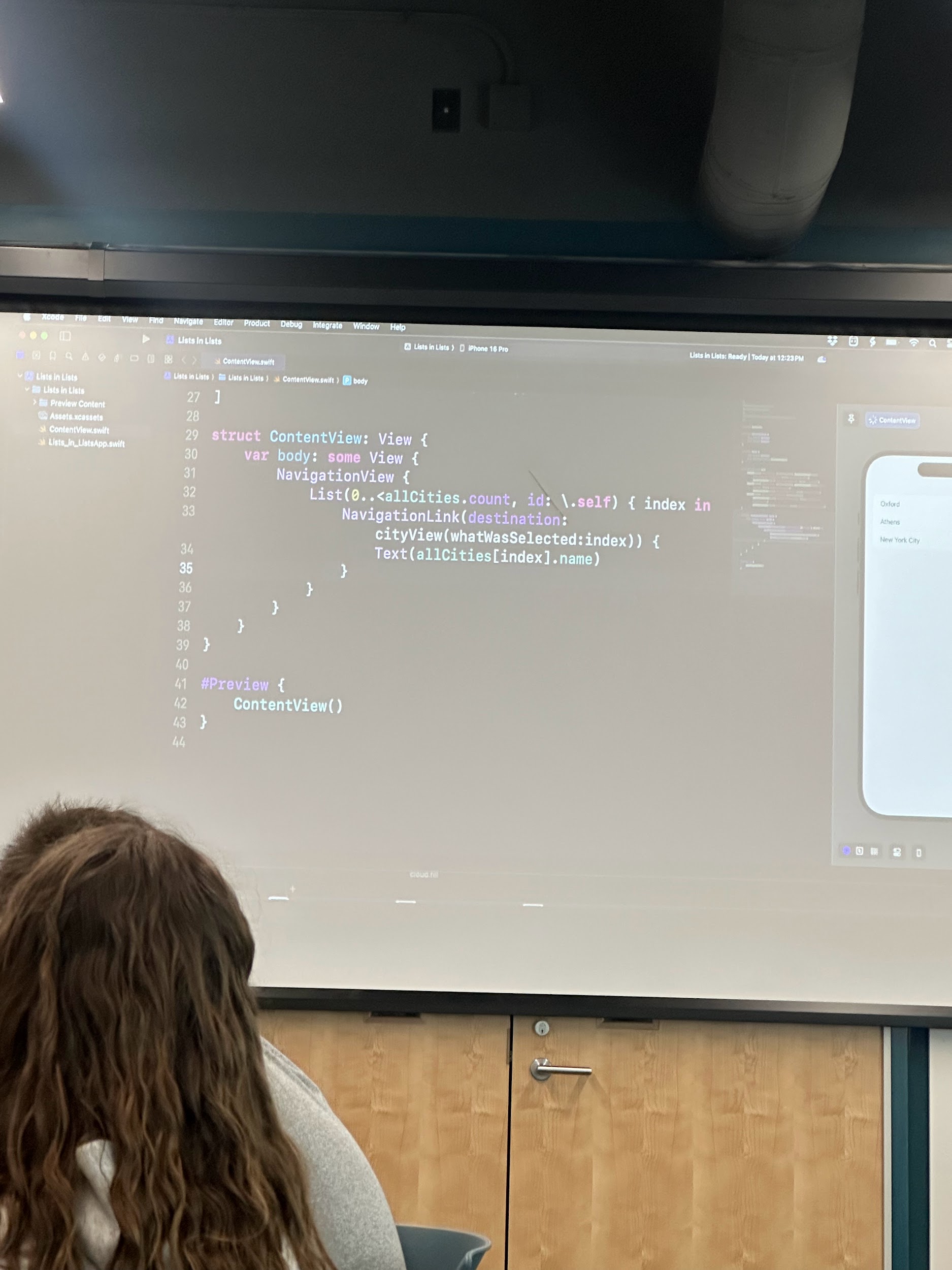
}

Var allCities = [

City(name: “Oxford”, attractions:

[Attraction(name: “miami university”, type:

“University” ),

CityView (not content or detail)

@state var CitySelection = 0

@state var attractionSelection = 0

Var body: some View {

Vstack

\*place other photo

**3/18**

**import** SwiftUI

**struct** Meal: Identifiable, Codable {

**var** id = UUID()

**var** name: String

**var** category: String

**var** image: String

**var** description: String

**var** isFavorite: Bool

}

**func** loadMeals() -> [Meal] {

**if** **let** savedData = UserDefaults.standard.data(forKey: "savedMeals") {

**if** **let** decoded = **try**? JSONDecoder().decode([Meal].**self**, from: savedData) {

**return** decoded

}

}

**return** []

}

**func** saveMeals(mealsToBeSaved: [Meal]) {

**if** **let** encoded = **try**? JSONEncoder().encode(mealsToBeSaved) {

UserDefaults.standard.set(encoded, forKey: "savedMeals")

}

}

**struct** ContentView: View {

@State **var** meals = loadMeals()

@State **private** **var** newMealName = ""

@State **private** **var** newMealCategory = ""

@State **private** **var** newMealDescription = ""

@State **private** **var** showAddMealView = **false**

@State **private** **var** searchText = ""

**var** filteredMeals: [Meal] {

**if** searchText.isEmpty {

**return** meals

} **else** {

**return** meals.filter { $0.name.localizedCaseInsensitiveContains(searchText) }

}

}

**var** body: **some** View {

NavigationView {

VStack {

Text("Meal Planner")

.font(.largeTitle)

.bold()

.padding()

TextField("Search Meals", text: $searchText)

.textFieldStyle(RoundedBorderTextFieldStyle())

.padding()

List(filteredMeals.indices, id: \ .**self**) { index **in**

NavigationLink(destination: SwiftUIView(selectedMeal: $meals[index])) {

HStack {

Image(meals[index].image)

.resizable()

.scaledToFit()

.frame(width: 90, height: 100)

.cornerRadius(10)

VStack(alignment: .leading) {

Text(meals[index].name)

.font(.title2)

.fontWeight(.semibold)

Text(meals[index].category)

.font(.headline)

.fontWeight(.light)

.foregroundColor(Color.gray)

}

}

}

}

.navigationTitle("Meal Planner")

Button("Add Meal") {

showAddMealView.toggle()

}

.padding()

.background(Color.blue)

.foregroundColor(.white)

.cornerRadius(10)

.sheet(isPresented: $showAddMealView) {

AddMealView(meals: $meals)

}

}

}

}

}

**struct** AddMealView: View {

@Binding **var** meals: [Meal]

@State **private** **var** name = ""

@State **private** **var** category = ""

@State **private** **var** description = ""

@Environment(\.presentationMode) **var** presentationMode

**var** body: **some** View {

NavigationView {

Form {

Section(header: Text("Meal Details")) {

TextField("Meal Name", text: $name)

TextField("Category", text: $category)

TextField("Description", text: $description)

}

}

.navigationBarItems(trailing: Button("Save") {

**let** newMeal = Meal(name: name, category: category, image: "default", description: description, isFavorite: **false**)

meals.append(newMeal)

saveMeals(mealsToBeSaved: meals)

presentationMode.wrappedValue.dismiss()

})

}

}

}

**4/15**

ENROLL IN APPLE DEVE

Request permission for notifications by running

Import UserNotifications

Var hasPermissionForAuthorization = false

Let userNotificationCenter = UNUserNotificationCenter.current()

func requestnotificationAuthorization() {

Let authOptions =

UNAuthorizationOptions.init(arrayLiteral: .Alert,

.badge, .)

Usernotificationcenter.requestAuthorization(options: authOptions) {

(sucess, error) in

If let error = error {

Has PermissionforAuthorization = false

Print (“there was an error”)

} else {

hasPermissionForAuthorization = true

}

}

}

Func createNotification() {

let notificationContent = UNMutableNotificationContent()

notificationContent. title = "End of Class"

notifica

tionContent.body = "IMS 351 ends at 1pm"

let trigger = UNTimeIntervalNotificationTrigger(timeInterval:

TimeInterval(10), repeats: false)

let identifier = "My Time Interval Notification"

let notificationQuery = UNNotificationRequest(identifier:

identifier, content: notificationContent, trigger: trigger)

userNotificationCenter.add (notificationQuery) { (error) in

if let error = error {

print("error in your request")

}

}

Check photos for lines 40-60