

# Programming Practices 1 Page Summary

## 1. MVVM Architecture (The Structure)

**Goal:** Separation of Concerns. The UI shouldn't know about API calls; the Logic shouldn't know about UI.

- **View (The Face):**
    - *Role:* Layout and appearance only.
    - *Action:* Listens to the ViewModel. Displays data.
  - **ViewModel (The Brain):**
    - *Role:* Holds the state (e.g., `isLoading`, `usersList`). Formats data for the View.
    - *Action:* Fetches data from Model, processes it, and publishes changes. **Never import UIKit/SwiftUI.**
  - **Model (The Data):**
    - *Role:* Raw data structures and business logic.
    - *Action:* Interacts with databases/services.
- 

## 2. Naming Conventions

Swift is a **CamelCase** language.

- **UpperCamelCase:** Classes, Structs, Enums, Protocols (`UserProfile`)
- **lowerCamelCase:** Variables, Functions, Constants (`userProfile`)

**The Verb Dictionary (Function Naming)**

- Working with Database: **Fetch, Delete**
  - Working with on-app data: **Get, Remove**
  - Other common terms: **Is, Update, Create**
- 

## 3. Documentation Practices

Comments explain **WHY**, Code explains **WHAT**.

- **Avoid:** Explaining syntax or obvious flow
- **Preferred:** Self-documenting variable names.
  - *Bad:* `if user.age > 18 { ... }`
  - *Good:* `let isLegalAdult = user.age >= 18; if isLegalAdult { ... }`
- **Preferred:** Contextual comments (`// Minimum tip is $5 to cover gas costs`).

**Formal Documentation (SwiftDoc)** Use `///` for shared functions.

Swift

`/// Calculates total cost.`

`/// - Parameter subtotal: The raw price.`

`/// - Returns: Price with tax included.`

`func calculateTotal(for subtotal: Double) -> Double { ... }`

**Navigation:** Use `// MARK: - Section Name` to organize large files.

---

## 4. Safety & Cleanliness

- **Utilize Visibility Modifiers:** If it is only going to be used internally, make it *private*!
- **No Force Unwrapping (!):** Avoid using `label.text!`. It causes crashes.
  - *Fix:* Use `if let` or `guard let` to handle nils gracefully.
- **No Magic Numbers:** Don't hardcode numbers in logic.
  - *Bad:* `if password.count > 8`
  - *Good:* `let minPasswordLength = 8; if password.count > minPasswordLength`