### **Lab 13-** **Create Button Label using Custom Style Class in Swift**

**Creating a Custom Style Class**

* To apply consistent styles across multiple views or view controllers, create a custom style class named as **Style**.

**Create a Custom Style Class:**

import UIKit

class Style {

static func applyPrimaryButtonStyle(to button: UIButton) {

button.tintColor = UIColor.white

button.backgroundColor = UIColor.systemBlue

button.layer.cornerRadius = 10

// Add shadow

button.layer.shadowColor = UIColor.black.cgColor

button.layer.shadowOpacity = 0.5

button.layer.shadowOffset = CGSize(width: 2, height: 2)

button.layer.shadowRadius = 5

}

static func applyPrimaryLabelStyle(to label: UILabel) {

label.textColor = UIColor.darkGray

label.font = UIFont.systemFont(ofSize: 24)

}

}

**Use the Custom Style Class:**

import UIKit

class ViewController: UIViewController {

override func viewDidLoad() {

super.viewDidLoad()

// View background color

self.view.backgroundColor = UIColor.lightGray

// Create a label

let label = UILabel(frame: CGRect(x: 20, y: 50, width: 300, height: 50))

label.text = "Hello, World!"

// Apply custom label style

Style.applyPrimaryLabelStyle(to: label)

// Create a button

let button = UIButton(type: .system)

button.frame = CGRect(x: 20, y: 120, width: 200, height: 50)

button.setTitle("Tap me", for: .normal)

// Apply custom button style

Style.applyPrimaryButtonStyle(to: button)

// Add views to the main view

self.view.addSubview(label)

self.view.addSubview(button)

// Add target-action for the button

button.addTarget(self, action: #selector(buttonTapped), for: .touchUpInside)

}

@objc func buttonTapped() {

print("Button was tapped!")

}

}

**Explanation**

**Custom Style Class:**

* Style.applyPrimaryButtonStyle(to:): Applies a custom style to a button.
* Style.applyPrimaryLabelStyle(to:): Applies a custom style to a label.

**Usage in ViewController:**

* Create a UILabel and a UIButton.
* Apply the custom styles using the Style class.
* Add the views to the main view and set up the button’s action.

**Summary**

Using custom styles and a dedicated style class in Swift makes it easy to maintain a consistent look and feel across your iOS app. This approach helps you avoid code duplication and makes your code more maintainable and readable.