### **Lab Exercise: Stack View Programmatically in iOS Swift**

**Objective:**

Learn how to create and use a stack view programmatically to arrange UI elements in an iOS app.

**Tasks:**

**Set Up the Project:**

* Create a new Xcode project named StackViewDemo.
* Choose "App" under iOS and set the language to Swift.

**Implement the View Controller:**

* Open ViewController.swift.
* Replace the contents of ViewController.swift with the following code:

import UIKit

class ViewController: UIViewController {

let stackView: UIStackView = {

let stackView = UIStackView()

stackView.axis = .vertical

stackView.spacing = 20

stackView.alignment = .center

stackView.translatesAutoresizingMaskIntoConstraints = false

return stackView

}()

let redView: UIView = {

let view = UIView()

view.backgroundColor = .red

view.translatesAutoresizingMaskIntoConstraints = false

view.widthAnchor.constraint(equalToConstant: 200).isActive = true

view.heightAnchor.constraint(equalToConstant: 100).isActive = true

return view

}()

let blueView: UIView = {

let view = UIView()

view.backgroundColor = .blue

view.translatesAutoresizingMaskIntoConstraints = false

view.widthAnchor.constraint(equalToConstant: 200).isActive = true

view.heightAnchor.constraint(equalToConstant: 100).isActive = true

return view

}()

override func viewDidLoad() {

super.viewDidLoad()

// Do any additional setup after loading the view.

// Add stack view to the view

view.addSubview(stackView)

// Add red view to the stack view

stackView.addArrangedSubview(redView)

// Add blue view to the stack view

stackView.addArrangedSubview(blueView)

// Configure layout constraints for stack view

NSLayoutConstraint.activate([

stackView.centerXAnchor.constraint(equalTo: view.centerXAnchor),

stackView.centerYAnchor.constraint(equalTo: view.centerYAnchor)

])

}

}

**Run the App:**

* Run the app on a simulator or a physical device.
* Verify that the red and blue views are arranged vertically in a stack view.
* Verify that the views are positioned correctly and resized according to the stack view's properties.

**Summary:**

This lab exercise provides hands-on experience with implementing a stack view programmatically in an iOS app using Swift. By completing the exercise, you will gain practical knowledge of using stack views to organize and layout UI elements dynamically in iOS apps.