### **Lab 10:** **Extension in Swift**

Extensions in Swift allow you to add new functionality to existing types, including types defined by the standard library or your own custom types. This lab exercise will cover extending various types with additional functionality.

**Part 1: Extending Standard Types**

1. **String Extension:**

* Extend the String type with a computed property isNotEmpty that returns true if the string is not empty.

extension String {

var isNotEmpty: Bool {

return !self.isEmpty

}

}

1. **Double Extension:**

* Define a simple extension to the Double type that adds a computed property to convert a value from kilometers to miles.

extension Double {

var kmToMiles: Double {

return self \* 0.621371

}

}

// Using the extension

let kilometers: Double = 5.0

print("\(kilometers) km is equal to \(kilometers.kmToMiles) miles.") // Output: 5.0 km is equal to

1. **String Extension:**

* Extend the String type to add a computed property that returns the reversed string.

extension String {

var reversedString: String {

return String(self.reversed())

}

}

// Using the extension

let originalString = "Hello"

print("Original: \(originalString), Reversed: \(originalString.reversedString)")

**Part 2: Extending Custom Types**

1. **Point Structure Extension:**

* Define a structure called Point with properties x and y.
* Extend the Point structure with a method distance(to:) that calculates the distance between two points.

struct Point {

var x: Double

var y: Double

}

extension Point {

func distance(to other: Point) -> Double {

let deltaX = other.x - self.x

let deltaY = other.y - self.y

return (deltaX \* deltaX + deltaY \* deltaY).squareRoot()

}

}

1. **Date Extension:**

* Extend the Date type with a computed property isPast that returns true if the date is in the past.

extension Date {

var isPast: Bool {

return self < Date()

}

}

**Part 3: Using Extensions**

1. **Using String Extension:**

* Create a string variable and check if it's not empty using the isNotEmpty property.

let text = "Hello, World!"

if text.isNotEmpty {

print("The string is not empty.")

} else {

print("The string is empty.")

}

1. **Using Point Structure Extension:**

* Create two instances of the Point structure and calculate the distance between them.

let point1 = Point(x: 1.0, y: 2.0)

let point2 = Point(x: 4.0, y: 6.0)

let distance = point1.distance(to: point2)

print("The distance between points is \(distance).")

1. **Using Date Extension:**

* Create a date and check if it's in the past using the isPast property.

let currentDate = Date()

if currentDate.isPast {

print("The date is in the past.")

} else {

print("The date is not in the past.")

}

**Summary**

This lab exercise covered the basics of using extensions in Swift, including extending standard types and custom types with additional functionality. By completing these tasks, you've learned how to enhance existing types with new properties, methods, and computed properties, promoting code reuse and clarity. Experiment with additional extensions to further explore the capabilities of Swift extensions.