

```
//  
// Conversions.swift  
// TemperatureConverter  
//  
// Created by Andrew Rozendal on 2018-02-20.  
// Copyright © 2018 Camosun. All rights reserved.  
//
```

```
import Foundation
```

```
class Conversions {  
    //Required Properties for all Conversions  
    let title: String  
    let leftButtonText: String  
    let rightButtonText: String  
    let leftButtonFunction: (Double) -> Double  
    let rightButtonFunction: (Double) -> Double  
  
    // Initializes a Conversions object with the passed parameters  
    init(title: String, leftButtonText: String, rightButtonText: String, leftButtonFunction:  
@escaping (Double) -> Double, rightButtonFunction: @escaping (Double) -> Double){  
        self.title = title  
        self.leftButtonText = leftButtonText  
        self.rightButtonText = rightButtonText  
        self.leftButtonFunction = leftButtonFunction  
        self.rightButtonFunction = rightButtonFunction  
    }  
}
```

```
// Temperature Conversions  
class TemperatureConversion: Conversions {  
  
    //Converts the passed C value to F  
    func convertCtoF(val: Double) -> Double {  
        return val * 9.0 / 5.0 + 32.0  
    }  
  
    //Converts the passed F value to C  
    func convertFtoC(val: Double) -> Double {  
        return (val - 32.0) * 5.0 / 9.0  
    }  
}
```

```
//Initialized by Conversions super class with Temperature specific labels
init(){
    super.init(title: "Temperature", leftButtonText: "C° to F°", rightButtonText: "F° to C°",
leftButtonFunction: convertCtoF, rightButtonFunction: convertFtoC)
}
}
```

// Area Conversions

```
class AreaConversion: Conversions {
```

```
    //Converts the passed Ha value to Ac
```

```
    func convertHectarestoAcres(val: Double) -> Double {
        return val * 2.471
    }
}
```

```
    //Converts the passed Ac value to Ha
```

```
    func convertAcrestoHectares(val: Double) -> Double {
        return val * 0.405
    }
}
```

```
//Initialized by Conversions super class with Area specific labels
```

```
init(){
    super.init(title: "Area", leftButtonText: "ha to ac", rightButtonText: "ac to ha",
leftButtonFunction: convertHectarestoAcres, rightButtonFunction:
convertAcrestoHectares)
}
}
```

// Length Conversions

```
class LengthConversion: Conversions {
```

```
    //Converts the passed ft value to m
```

```
    func convertFeettoMetres(val: Double) -> Double {
        return val * 0.305
    }
}
```

```
    //Converts the passed m value to ft
```

```
    func convertMetrestoFeet(val: Double) -> Double {
        return val * 3.281
    }
}
```

```
//Initialized by Conversions super class with Length specific labels
init(){
    super.init(title: "Length", leftButtonText: "ft to m", rightButtonText: "m to ft",
leftButtonFunction: convertFeettoMetres, rightButtonFunction: convertMetrestoFeet)
}
}
```

// Weight Conversions

```
class WeightConversion: Conversions {
```

```
    //Converts the passed lbs value to kg
```

```
    func convertPoundstoKilos(val: Double) -> Double {
        return val * 0.454
    }
```

```
    //Converts the passed kg value to lbs
```

```
    func convertKilostoPounds(val: Double) -> Double {
        return val * 2.205
    }
```

```
//Initialized by Conversions super class with Weight specific labels
```

```
init(){
    super.init(title: "Weight", leftButtonText: "kg to lbs", rightButtonText: "lbs to kg",
leftButtonFunction: convertKilostoPounds, rightButtonFunction: convertPoundstoKilos)
}
}
```

```

//
// UnitConversionViewController.swift
// TemperatureConverter
//
// Created by Andrew Rozendal on 2018-02-19.
// Copyright © 2018 Camosun. All rights reserved.
//

import UIKit
import os

class UnitConversionViewController: UITableViewController {
    //MARK: Properties

    // Initialize all Conversion sub-classes we want to provide
    let cellIdentifier = "conversionType"
    let conversions = [TemperatureConversion(), AreaConversion(), LengthConversion(),
WeightConversion()]

    //MARK: Delegate functions
    override func viewDidLoad() {
        super.viewDidLoad()
    }

    override func didReceiveMemoryWarning() {
        super.didReceiveMemoryWarning()
        // Dispose of any resources that can be recreated.
    }

    override func tableView(_ tableView: UITableView, numberOfRowsInSection section:
Int) -> Int {
        // return the number of rows
        return conversions.count
    }

    override func tableView(_ tableView: UITableView, cellForRowAt indexPath:
IndexPath) -> UITableViewCell {
        guard let cell = tableView.dequeueReusableCell(withIdentifier: cellIdentifier, for:
indexPath) as? UnitConversionTableViewCell else {
            // selected cell was not a UnitConversionTableViewCell
            fatalError("Selected cell is not of type \(cellIdentifier)")
        }
    }

```

```

    guard let label = cell.textLabel else {
        // selected cell did not have a textLabel
        fatalError("Selected cell does not have textLabel")
    }

    // Configure the cell...
    label.text = conversions[indexPath.item].title
    return cell
}

// MARK: - Navigation

// Prepares the Conversion page the app is about to navigate to with the relevant
conversion
// object which contains the required logic
override func prepare(for segue: UIStoryboardSegue, sender: Any?) {
    super.prepare(for: segue, sender: sender)

    guard let selectedCell = sender as? UnitConversionTableViewCell else {
        // The destination is not a UnitConversionTableViewCell
        fatalError("Unexpected destination \(String(describing: sender))")
    }

    guard let cellLabelText = selectedCell.textLabel?.text else {
        // The text on the textLabel of the selected cell is nil, or the selected cell has no
        textLabel
        fatalError("Selected cell has no label with text")
    }

    var conversionIndex: Int?

    // Find the matching Conversion instance in the Conversions list
    for i in 0 ..< conversions.count {
        if conversions[i].title == cellLabelText {
            conversionIndex = i
            break
        }
    }

    guard let i = conversionIndex else {
        // conversion index was never initialized

```

```
        os_log("Conversion Index was not set properly", log: OSLog.default, type:
.debug)
        return
    }

    guard let destination = segue.destination as? ViewController else {
        // Destination was unable to be cast as ViewController
        fatalError("Unexpected destination \(segue.destination)")
    }

    destination.currentConversion = conversions[i]
}
}
```

```

//
// ViewController.swift
// TemperatureConverter
//
// Created by Andrew Rozendal on 2018-02-16.
// Copyright © 2018 Camosun. All rights reserved.
//

import UIKit
import os

class ViewController: UIViewController {
    //MARK: Properties

    // The instance of Conversions to use for the current view
    var currentConversion: Conversions? = nil
    // The name of the current conversion
    @IBOutlet weak var conversionTitle: UILabel!
    // Identifies user conversion choice
    @IBOutlet weak var desiredConversionChoice: UISegmentedControl!
    // Field for user value to convert
    @IBOutlet weak var valueToConvertField: UITextField!
    // Label to hold end result
    @IBOutlet weak var resultField: UILabel!

    //Mark: Delegate Methods
    override func viewDidLoad() {
        super.viewDidLoad()

        // Set labels for the view based on the conversion instance
        guard let c = currentConversion else {
            os_log("Cannot grab attributes from a Conversions item that is nil", log:
OSLog.default, type: .debug)
            return
        }

        self.conversionTitle.text = c.title
        self.desiredConversionChoice.setTitle(c.leftButtonText, forSegmentAt: 0)
        self.desiredConversionChoice.setTitle(c.rightButtonText, forSegmentAt: 1)
    }

    override func didReceiveMemoryWarning() {
        super.didReceiveMemoryWarning()
    }

```

```

        // Dispose of any resources that can be recreated.
    }

    //MARK: Actions

    // Called when convert button pressed
    // Converts the value in the input field according to the formula associated with the
    segment selection.
    // Outputs the result to the result field
    @IBAction func convertValueBtn(_ sender: Any) {
        guard let textValue = self.valueToConvertField.text else {
            // valueToConvertField was nil
            self.resultField.text = "N/A"
            return
        }

        guard let value = Double(textValue) else {
            // textValue could not be converted to Double
            self.resultField.text = "N/A"
            return
        }

        guard let c = currentConversion else {
            // Conversion object was nil
            os_log("Cannot grab attributes from a Conversions item that is nil", log:
OSLog.default, type: .debug)
            return
        }

        if self.desiredConversionChoice.selectedSegmentIndex == 0 {
            // Left button conversion
            self.resultField.text = String((c.leftButtonFunction)(value))
        } else {
            // Right button conversion
            self.resultField.text = String((c.rightButtonFunction)(value))
        }
    }
}

```