

```
//  
// ItemCell.swift  
// Homepowner2  
//  
// Created by Chelsea Irizarry on 11/11/16.  
// Copyright © 2016 Chelsea Irizarry. All rights reserved.  
//
```

```
import UIKit
```

```
class ItemCell: UITableViewCell {  
  
    @IBOutlet var nameLabel: UILabel!  
    @IBOutlet var serialNumberLabel: UILabel!  
    @IBOutlet var valueLabel: UILabel!  
  
    func updateLabels() {  
        let bodyFont = UIFont.preferredFont(forTextStyle:  
UIFontTextStyle.body)  
        nameLabel.font = bodyFont  
        valueLabel.font = bodyFont  
  
        let caption1Font = UIFont.preferredFont(forTextStyle:  
UIFontTextStyle.caption1)  
        serialNumberLabel.font = caption1Font  
    }  
}
```

```
//  
// ItemsViewController.swift  
// Homepowner2  
//  
// Created by Chelsea Irizarry on 10/20/16.  
// Copyright © 2016 Chelsea Irizarry. All rights reserved.  
//
```

```
import UIKit
```

```
//Define a UITableViewController subclass named ItemsViewController  
class ItemsViewController: UITableViewController {
```

```
    var itemStore: ItemStore!
```

```
    //Adding Rows
```

```
    @IBAction func addNewItem(sender: AnyObject){
```

```
        //Create a new item and add it to the store
```

```
        let newItem = itemStore.createItem()
```

```

        //Figure out where that item is in the array
        if let index = itemStore.allItems.index(of: newItem) {
            let indexPath = NSIndexPath(row: index, section: 0)

            //Insert this new row into the table
            tableView.insertRows(at: [indexPath as IndexPath],
with: .automatic)

        }
    }

    //Deleting Rows
    override func tableView(_ tableView: UITableView, commit
editingStyle: UITableViewCellEditingStyle, forRowAt indexPath:
IndexPath) {

        //If the table view is asking to commit a delete command...
        if editingStyle == .delete {
            let item = itemStore.allItems[indexPath.row]

            let title = "Delete \ \(item.name)?"
            let message = "Are you sure you want to delete this item?"

            let ac = UIAlertController(title: title, message: message,
preferredStyle: .actionSheet)

            let cancelAction = UIAlertAction(title: "Cancel",
style: .cancel, handler: nil)
            ac.addAction(cancelAction)

            let deleteAction = UIAlertAction(title: "Delete",
style: .destructive, handler: { (action: UIAlertAction) -> Void in
                //Remove the item from the store
                self.itemStore.removeItem(item: item)

                //Also remove that row from the table view with an
animation
                self.tableView.deleteRows(at: [indexPath],
with: .automatic)
            })
            ac.addAction(deleteAction)

            //Present the alert controller
            present(ac, animated: true, completion: nil)

        }
    }

    //Moving Rows

```

```

    override func tableView(_ tableView: UITableView, moveRowAt
sourceIndexPath: IndexPath, to destinationIndexPath: IndexPath) {
        //Update the model
        itemStore.moveItemAtIndex(fromIndex: sourceIndexPath.row,
toIndex: destinationIndexPath.row)
    }

    //Editing Mode
    @IBAction func toggleEditMode(sender: AnyObject){
        //If your are currently in editing mode...
        if isEditing {
            //Change text of button to inform user of state
            sender.setTitle("Edit", for: .normal)

            //Turn off editing mode
            setEditing(false, animated: true)
        }

        else{
            //Change text of button to inform user of state
            sender.setTitle("Done", for: .normal)

            //Enter editing mode
            setEditing(true, animated: true)
        }
    }

    override func tableView(_ tableView: UITableView,
numberOfRowsInSection section: Int) -> Int {
        return itemStore.allItems.count
    }

    override func tableView(_ tableView: UITableView, cellForRowAt
indexPath: IndexPath) -> UITableViewCell {
        //Get a new or recycled cell
        let cell = tableView.dequeueReusableCell(withIdentifier:
"ItemCell", for: indexPath) as! ItemCell

        //Update the labels for the new preferred text size
        cell.updateLabels()

        //Set the tex on the cell with the description of the item
        //that is at the nth index of items, where n = row this cell
        //will appear in on the tableview
        let item = itemStore.allItems[indexPath.row]

        //Configure the cell with the Item
        cell.nameLabel.text = item.name
        cell.serialNumberLabel.text = item.serialNumber
        cell.valueLabel.text = "$\(item.valueInDollars)"
    }

```

```
        return cell
    }

    override func viewDidLoad() {
        super.viewDidLoad()

        //Get the height of the status bar
        let statusBarHeight =
UIApplication.shared.statusBarFrame.height
        let insets = UIEdgeInsets(top: statusBarHeight, left: 0,
bottom: 0, right: 0)
        tableView.contentInset = insets
        tableView.scrollIndicatorInsets = insets

        tableView.rowHeight = UITableViewAutomaticDimension
        tableView.estimatedRowHeight = 65
    }
}
```