# TableDetailViews

1. Start with a copy of our previous TableP project. We had constructed table using property list in TableP during our class 6. Build and run to ensure that you have good fresh copy.
2. Add a new class file by highlighting TableP folder -> right click -> New File. Under iOS/Source pick Cocoa Touch Class. Provide TableDetailViewController as class Name and keep default of UIViewController for Subclass of Radio Button

3. Update tableDetail.h as follows:

```
@interface tableDetailViewController : UIViewController
@property (strong, nonatomic) NSString *inLabelText;
@property (nonatomic) NSInteger inRow;
@property (nonatomic) NSInteger inSection;

@property (strong, nonatomic) IBOutlet UILabel *outLabel;
@property (strong, nonatomic) IBOutlet UILabel *showRow;
@property (strong, nonatomic) IBOutlet UILabel *showSection;
```

4. Update tableDetail.m as follows:

```
@synthesize inLabelText, inRow, inSection;
@synthesize outLabel, showRow, showSection;
```

4. In story board add UIViewController. Add three labels. Make them as wide as you can. Set tableDetail as Custom Class for UIViewController that we just added (from Identity Inspector). From view controller connect three outlets for three labels.

5. Update tableViewController.h as follows:

to class var

```
IBOutlet UITableView *myTableView;
```

add class property

```
@property (nonatomic, retain) IBOutlet UITableView *myTableView;
```

6. Update story board as follows:

Connect out new myTableView outlet for view controlled by tablepViewController.

# TableDetailViews

Add a Navigation control to first table view. Editor -> Embed In -> Navigation Controller

From ViewController icon of our first table view, control drag to our detail view - second view controller that we just added, pick “Push” Segue as we are pushing this new view in.

Finally high light segue and goto attributes inspector. Provide “tableDetail” as segue name for the segue.

7. Update tableViewController.m as follows:

Import header for the new class we added
#import “tableDetail.h”

synthesize property
@synthesize myTableView;

Add two methods to segue

```
#pragma mark Segues
- (void)tableView:(UITableView *)tableView didSelectRowAtIndexPath:
(NSIndexPath *)indexPath {

    [self performSegueWithIdentifier:@"tableDetail" sender:self];

}

- (void) prepareForSegue:(UIStoryboardSegue *)segue sender:(id)sender
{
    if ([[segue identifier] isEqualToString:@"tableDetail"]) {

        tableDetail *detailVC = [segue destinationViewController];

        // index path to row that's selected - you don't need to store
row, or
        // section in global .....

        NSIndexPath *myIndexPath = [self.myTableView
indexPathForSelectedRow];

        // Get Row Number
        NSString *myFaculty = [self.faculty objectAtIndex:[myIndexPath
section]];
        NSArray *facultyCourses = [self.cod objectForKey:myFaculty];
        NSInteger row = [myIndexPath row];
        NSInteger section = [myIndexPath section];
```

# TableDetailViews

```
        // Pass parameters to VC
        detailVC.inLabelText = [facultyCourses objectAtIndex:
[myIndexPath row]];
        detailVC.inRow = row;
        detailVC.inSection = section;


    }
}
```

6. Build and run to verify that you can go from table to a second view where you can see our labels.

7. update tableDetail.m as follows:

```
- (void)viewDidLoad
{
    [super viewDidLoad];
    outLabel.text = self.inLabelText;
    showRow.text = [NSString stringWithFormat:@"Row passed to this VC
is %i", inRow];
    showSection.text = [NSString stringWithFormat:@"Section passed to
this VC is %i", inSection];

}
```

Build and Run - watch all the variables that are passed to tableDetail VC.