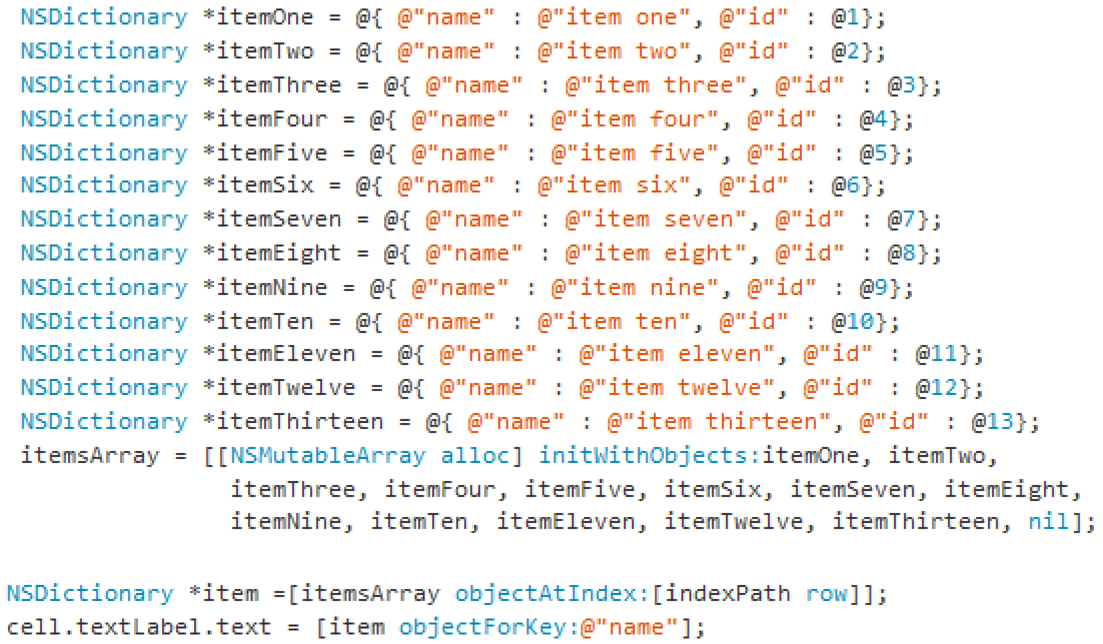
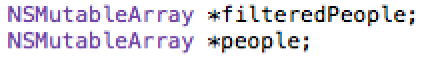
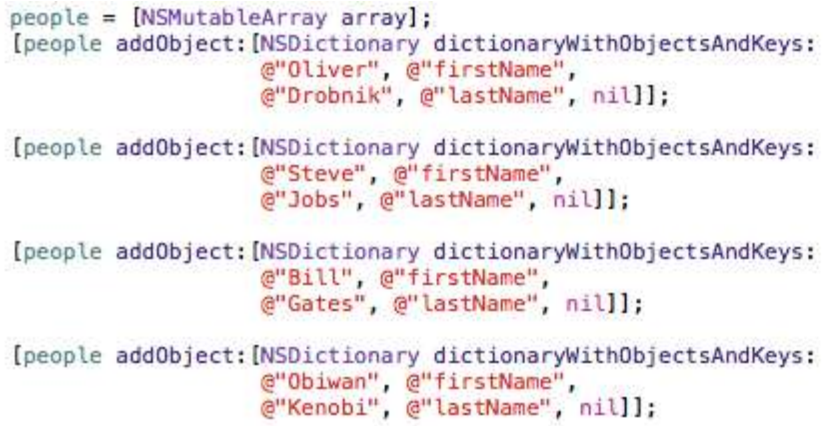


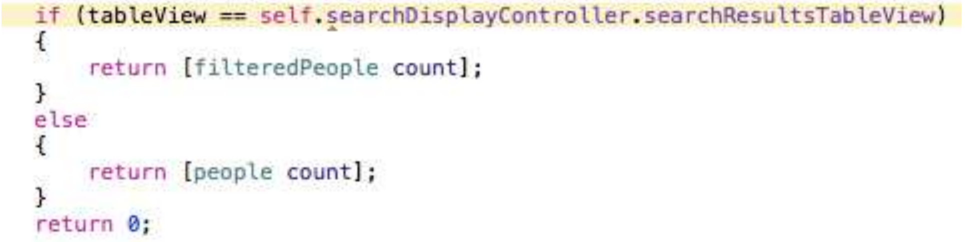
```
 NSDictionary *itemOne = @{ @"name" : @"item one", @"id" : @1};
 NSDictionary *itemTwo = @{ @"name" : @"item two", @"id" : @2};
 NSDictionary *itemThree = @{ @"name" : @"item three", @"id" : @3};
 NSDictionary *itemFour = @{ @"name" : @"item four", @"id" : @4};
 NSDictionary *itemFive = @{ @"name" : @"item five", @"id" : @5};
 NSDictionary *itemSix = @{ @"name" : @"item six", @"id" : @6};
 NSDictionary *itemSeven = @{ @"name" : @"item seven", @"id" : @7};
 NSDictionary *itemEight = @{ @"name" : @"item eight", @"id" : @8};
 NSDictionary *itemNine = @{ @"name" : @"item nine", @"id" : @9};
 NSDictionary *itemTen = @{ @"name" : @"item ten", @"id" : @10};
 NSDictionary *itemEleven = @{ @"name" : @"item eleven", @"id" : @11};
 NSDictionary *itemTwelve = @{ @"name" : @"item twelve", @"id" : @12};
 NSDictionary *itemThirteen = @{ @"name" : @"item thirteen", @"id" : @13};
 itemsArray = [[NSMutableArray alloc] initWithObjects:itemOne, itemTwo,
               itemThree, itemFour, itemFive, itemSix, itemSeven, itemEight,
               itemNine, itemTen, itemEleven, itemTwelve, itemThirteen, nil];

NSDictionary *item =[itemsArray objectAtIndex:[indexPath row]];
cell.textLabel.text = [item objectForKey:@"name"];
```



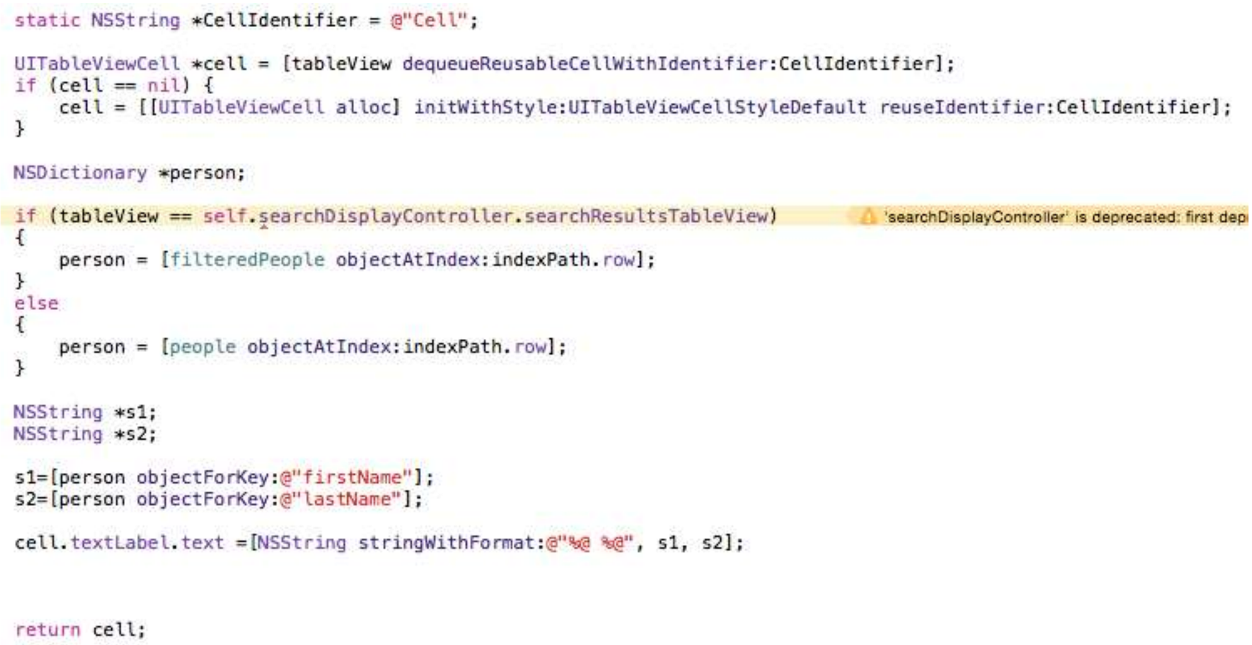
```
NSMutableArray *filteredPeople;
NSMutableArray *people;
```



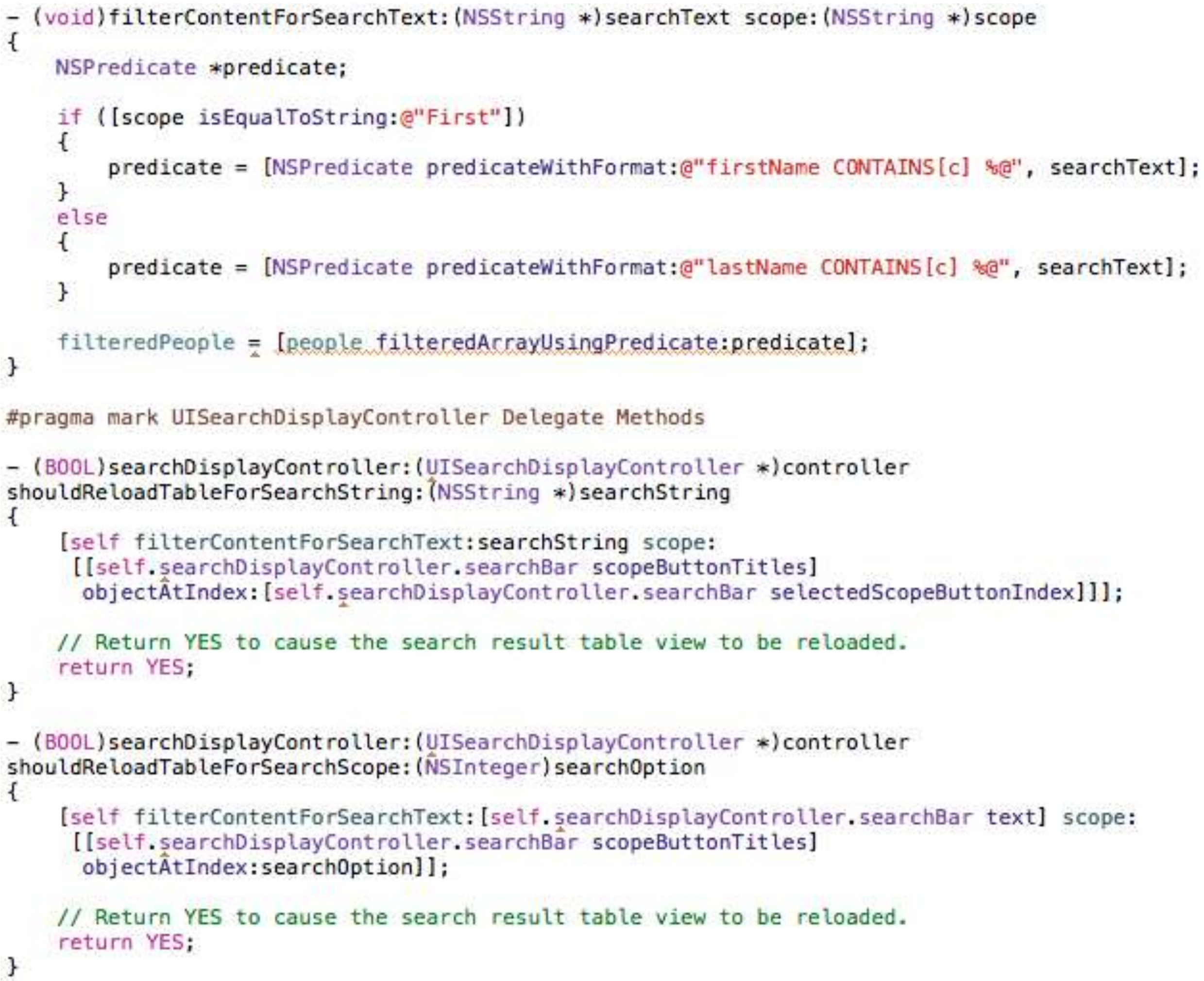
```
 people = [NSMutableArray array];
 [people addObject:[NSDictionary dictionaryWithObjectsAndKeys:
                   @"Oliver", @"firstName",
                   @"Drobnik", @"lastName", nil]];

 [people addObject:[NSDictionary dictionaryWithObjectsAndKeys:
                   @"Steve", @"firstName",
                   @"Jobs", @"lastName", nil]];

 [people addObject:[NSDictionary dictionaryWithObjectsAndKeys:
                   @"Bill", @"firstName",
                   @"Gates", @"lastName", nil]];

 [people addObject:[NSDictionary dictionaryWithObjectsAndKeys:
                   @"Obiwan", @"firstName",
                   @"Kenobi", @"lastName", nil]];
```



```
  if (tableView == self.searchDisplayController.searchResultsTableView)
  {
      return [filteredPeople count];
  }
  else
  {
      return [people count];
  }
  return 0;
```



```
static NSString *CellIdentifier = @"Cell";

UITableViewCell *cell = [tableView dequeueReusableCellWithIdentifier:CellIdentifier];
if (cell == nil) {
    cell = [[UITableViewCell alloc] initWithStyle:UITableViewCellStyleDefault reuseIdentifier:CellIdentifier];
}

NSDictionary *person;

if (tableView == self.searchDisplayController.searchResultsTableView)          'searchDisplayController' is deprecated: first dep
{
    person = [filteredPeople objectAtIndex:indexPath.row];
}
else
{
    person = [people objectAtIndex:indexPath.row];
}

NSString *s1;
NSString *s2;

s1=[person objectForKey:@"firstName"];
s2=[person objectForKey:@"lastName"];

cell.textLabel.text =[NSString stringWithFormat:@"%@ %@", s1, s2];


return cell;
```



```
- (void)filterContentForSearchText:(NSString *)searchText scope:(NSString *)scope
{
    NSPredicate *predicate;

    if ([scope isEqualToString:@"First"])
    {
        predicate = [NSPredicate predicateWithFormat:@"firstName CONTAINS[c] %@", searchText];
    }
    else
    {
        predicate = [NSPredicate predicateWithFormat:@"lastName CONTAINS[c] %@", searchText];
    }

    filteredPeople = [people filteredArrayUsingPredicate:predicate];
}

#pragma mark UISearchDisplayController Delegate Methods

- (BOOL)searchDisplayController:(UISearchDisplayController *)controller
shouldReloadTableForSearchString:(NSString *)searchString
{
    [self filterContentForSearchText:searchString scope:
     [[self.searchDisplayController.searchBar scopeButtonTitles]
      objectAtIndex:[self.searchDisplayController.searchBar selectedScopeButtonIndex]]];

    // Return YES to cause the search result table view to be reloaded.
    return YES;
}

- (BOOL)searchDisplayController:(UISearchDisplayController *)controller
shouldReloadTableForSearchScope:(NSInteger)searchOption
{
    [self filterContentForSearchText:[self.searchDisplayController.searchBar text] scope:
     [[self.searchDisplayController.searchBar scopeButtonTitles]
      objectAtIndex:searchOption]];

    // Return YES to cause the search result table view to be reloaded.
    return YES;
}
```