**Topic: Table View Controller**

Intro

* Look at due dates
* Reminder about code reviews

TableView Applications

* Basic Table
  + Look at basic concepts in the Xamarin Guide, Working with Tables and Cells, Part 1 – Table Parts and Functionality  
    <https://developer.xamarin.com/guides/ios/user_interface/tables/part_1_-_table_parts_and_functionality>
  + Walk-throughs:
    - Code from the Xamarin Guide, Part 2, **BasicTable**  
      <https://github.com/xamarin/ios-samples/tree/master/WorkingWithTables>
      * List of plants, no storyboard
      * Note: Data source doesn’t require an array, but it’s being converted List<string> users .ToArray(). Maybe programmer preference?
    - Look at the Ch. 8, **SimpleTable** – list of Dwarves (no index or detail view)
      * Has a TableViewController on the storyboard (different from the book, which has a ViewController on the storyboard)
      * Has detail text in the row
      * Has an images (star) that changes when the row is clicked
* Table with sections and index
  + Look at concepts in the Xamarin Guide, Part 2 – Populating a Table with Data
  + Walk-throughs:
    - Code from the Xamarin Guide, Part 2, B**asicTableIndex**
      * Reads list of plants from .txt file, no storyboard
      * Builds sections and an index from the data by using a loop and putting them into a dictionary of lists.
    - My dwarf example with an index: <https://github.com/LCC-CIT/cs235imTableViewDemo>

Additional references

* + Xamarin/ios-samples/PopulatingATable: Storyboard just has a ViewController + View.
    - Essentially the same as BasicTable, but with a storyboard?
    - Clicking on a row pops up an alert with row content
    - Bug! Threw an Objective-C run-time exception on the first try, then ran.
  + Xamarin/ios-samples/TableParts: Shows the parts of a table—rows, sections, section header and footer. (we didn’t use this one, but might want to next time)  
    <https://github.com/xamarin/ios-samples/tree/master/TableParts>
    - Changes to the tasks are stored back in the TableViewController’s data source.