1. Give two advantages and two disadvantages of using Xcode's storyboard approach over using Android Studio's UI Designer with editable layout xml files. You may consider ease-of-use, relative efficiency, and flexibility for controlling what is done in code versus what is done in the designer, and accessibility to the internal representation of the UI controls.

2. Explain the different data control approaches used by Java and Swift – garbage collection versus automatic reference counting (ARC). Consider ease of use for the app developer, and efficiency from both a time and space perspective. Does the Swift runtime for MacOSX applications (as opposed to iOS applications) use garbage collection? Look up the Android Runtime (ART), which is used in place of Dalvik Runtime Environment since API version 4.4. Explain the differences between these two runtime environments. Does this change the comparison of Android to iOS when it comes to execution efficient of apps (after they are installed)?

Discuss these differences between the Java approach and the Swift iOS approach (garbage collection versus Automatic Reference counting) in the context of the following code construction (assume the appropriate syntax for each language):

```java
String myString = “1-The quick brown fox jumped over the lazy dog”;
for ( int i= 0; i< 10000; i++){
    myString = new String (“I am string “+ i);
}
```

3. iOS Navigation Controllers. Explain the difference between the Tab Bar and the Navigation Bar in iOS apps. Name one Apple provided app that comes pre-installed on the iOS simulator that uses the Navigation Bar, and another one that uses the Tab Bar.

4. What is a segue in iOS? Describe how its created and used in Xcode's interface builder. Explain why a controller in an iOS app may want to implement the method:

```swift
override func prepareForSegue(segue: UIStoryboardSegue, sender: AnyObject?)
```

5. Explain the differences between UITableView in iOS and ExpandableListView with Android. What are the strengths and drawbacks of each?

6. Explain what the UITableViewDataSource delegate method shown below does. Why does the idiom for implementing this method include removing a table cell from a pool of cells and then checking whether the returned cell is nil?

```swift
override func tableView(tableView: UITableView, cellForRowAtIndexPath indexPath: NSIndexPath) -> UITableViewCell
```

7. Explain with a sentence each, the functionality of the UIPickerView delegate methods that we've implemented in all examples using a Picker.