Exam 1 Study Guide

A few study and test-taking hints:

1. Read over **all** the keynotes carefully. Most of the questions will come from there.
2. A few simple exam-taking tips:
    1. Answer a question if you can, but **use your time wisely**: if you don't know the answer after a few moments reflection, move on.
    2. Allocate time based on how many points the question is worth. Your goal is to optimize your overall score. If you don't know the answer to a 2 point question, **move on**.
3. Do not write more than you need to. For instance, if the question says:
    1. "What keyword does **x**?" **just write the keyword** that does x, not a novella.
    2. write a stored property, **just write the stored property,** not the enclosing class.
    3. assume you have a stored property -- you **do**! Do not rewrite it, just use it.
4. Do not answer a question twice, using "or" to distinguish between your two answers. The grader will grade the first answer.
5. Bring your own pencil(s), pen(s) and eraser(s). Seriously, it's a paper-based exam, people, why would you *not* bring these??
6. While exam coverage focuses on material since the last exam, you are still expected to be able to write basic Swift code.
7. You may bring a 8.5x11x0.0 sheet of paper with hand-written notes

Topics:

1. Life without storyboards
2. AutoLayout
3. TabBarControllers
4. Table Views
5. Table View Controllers
6. Closures
7. Navigation Controllers
8. Segues
9. MBaaS
10. Strings

Life Without Storyboards

1. True or false: an app resides in a window that occupies the entire screen
2. What is the purpose of a window's rootViewController?
3. How do you make a ViewController in code?
4. How do you make a UILabel in code, and give its text a particular value?
5. What does loadView() do? viewDidLoad()? viewWillAppear? When are they called, and who calls them?
6. What is a ViewController's **view** property good for?
7. What does addSubview() do?
8. Give 2 reasons why you might **not** want to use storyboards

Autolayout

1. What information establishes a view's frame?
2. What is the relationship between a view's frame and its constraints?
3. What is a constraint?
4. What do dashed orange lines represent on storyboard?
5. What are meant by ambiguous constraints?
6. What are meant by satisfiable constraints? Unsatisfiable constraints?
7. What are prototype constraints? How do they work?

Tab Bar Controllers

1. What is contained in a tab bar controller's **viewControllers** property?
2. What type is viewControllers?
3. Name the 3 properties in a UITabBarItem, including their types.
4. Assign a title "Stats", and image (named "Stats.jpg" to a view controller's tabBarItem property. Do this both in code and in storyboard.
5. Which class defines **tabBarItem** as a property?
6. If you assign a title and image to a tabBarItem in code, why can it **not** be done in viewDidLoad() or viewWillAppear()?
7. True or false: every view controller has a tabBarController property, but it only becomes relevant when that view controller is in a tabbed application
8. In storyboard, how do you add a view controller to a tab bar controller?
9. What is the point of making a property in the AppDelegate static?
10. Can a static property be changed?
11. What is the purpose of a singleton?
12.

Table Views

1. Where does a table view get its data from?
2. Give the signatures of the 3 key methods in UITableViewDataSource
3. Name a method, found in the UITableViewDelegate method, that is invoked when a user taps on a table view cell.
4. How do you designate a UIViewController subclass as a UITableViewDataSource or UITableViewDelegate in storyboard?
5. What does dequeueReusableCell(withIdentifier: for:) do?
6. What is the purpose of a table view cell's reuse identifier?
7. What is the appearance of a cell if it is Basic, Right Detail, Left Detail, or Subtitle?
8. How do you make multiple sections in a table view?
9. Do you specify multiple sections in code or storyboard?
10. How do you specify different cells for different sections?
11. How do you specify that different sections have different numbers of cells.
12. What is a custom cell type?
13. How do you access labels (or other views) in a custom cell?
14. What does viewWithTag(:_) do? What does it return? Do you need to use casting with it?
15. Practical problem: Given an array of structs, where the structs have multiple properties populate a table view with those structs. (e.g., see the Restaurant example in your last assignment).

16. Practical problem: a custom cell has 2 UILabels and a UIImageView, tagged 10, 20 and 30. Write code to populate them with values "I am a label", "I am another label", and the contents of "journey.jpg".

Table View Controllers

1. What is a table view controller?
2. What protocols does it implement?
3. What is its view look like, and how is it constrained?
4. If a view required a table view by itself, could it subclass UITableViewController? What if the view required a table view and a button or text field?
5. Describe the advantages of a UITableViewController over a UITableView + UIViewController

Closures

1. What is meant by a first-class type?
2. What is a closure? How you write one?
3. Describe the syntax of a closure.
4. How do you write a function that can be passed in a closure?
5. What shortcuts are available when defining closures?
6. Is {(x:Int) -> Bool in return x % 2 == 0 } a legitimate closure?
7. What could you leave out of the closure and still be syntactically correct.
8. Write a method, **useIt()** with 3 parameters -- **start** and **stop** (Ints) and **okToPrint**, that can be passed in a closure of the form shown in question 6. It will print out all the numbers from start to stop (exclusive) for which okToPrint() returns true. For example, useIt(start:0, stop: 24, okToPrint:{(x:Int) -> Bool in return x % 2 == 0 }) would print out 0, 2, 4, 6, ..., 22
9. Write a method, loseIt(min:max:data:check:), that will be passed in two Doubles, min and max, an array, data, and a closure, check(), that accepts two Doubles and returns a Bool. loseIt() will print out the elements of data, between min and max inclusive, that satisfy check (that is, for which check returns true).
10. Explain how map() and filter() work.
11. Given an array of Ints, **data**, write a map() statement to return an array of Bools, true for each even element of data, false for each odd element
12. Given an array of ints, **data**, write a filter statement to return just those elements that are divisible by 5.

Navigation Controllers

1. What data structure does a navigation controller use to store its view controllers?
2. What is a rootViewController? When is it visible?
3. Name 3 properties in a UINavigationItem.
4. Which class defines **navigationItem** as a property?
5. True or false: every view controller has a property, navigationController, that becomes relevant when that view controller is contained in a navigation controller
6. What do pushViewController(_:animated:) and popViewController(animated:) do?

Segues

1. What is a segue?
2. What objects can a user tap on to trigger a segue?
3. How do you create a segue in storyboard?
4. Give the signature of the method that gets invoked as a segue is about to occur
5. What is a segue's identifier? When is it needed?
6. A user taps on a button on view controller **AVC**, to invoke a segue to view controller **BVC**. In prepare(for segue:sender:), what is segue.sourceViewController? segue.destinationViewController?
7. Explain what is meant by a presentation
8. What is an unwind segue? How do you create one in storyboard? To unwind to a view controller, what method would need to be in that view controller?
9. When calling an unwind segue, does prepare(for segue:sender:) get called? In which view controller, the source or destination?

MBaaS

1. What does MBaaS stand for?
2. To use a class in Backendless:
    1. what class must it subclass?
    2. what initializer must it define?
    3. what properties must be non-optional?
    4. how many properties must it define?
3. What does a class's objectId represent?
4. What, on Backendless 'servers, corresponds to an IDataStore?
5. Given an IDataStore cityDataStore, what does cityDataStore.save() do?
6. Explain the difference between synchronous and asynchronous method calls. Which one gets control back to the user faster?
7. Does Backendless use closures? Why?
8. What is the purpose of Types.tryblock(:catchblock:) ?
9. If you want to update an object's values, how would you do that in Backendless?
10. In a 1:n relationship, would you use addRelation() or setRelation() to establish the relationship between a parent and children
11. In code, what property needs to be defined in a parent class to establish a 1:n relationship with children? What will its type be?
12. True or false: the NotificationCenter can be used to post notifications, and establish methods to be invoked when a notification is posted.
13. What does this statement do: NotificationCenter.default.post(name: .CitiesReloaded, object:nil)
14. What does this statement do: NotificationCenter.default.addObserver(self, selector:#selector(handleThis), name:.CitiesReloaded, object:nil)
15. What are the capabilities of Backendless' User Service?
16. Describe the 3 different permission categories in Backendless.
17.

Dictionaries

1. What is the purpose of a Dictionary? What is the equivalent in Java?
2. Declare a dictionary that has a key of String, and a value of Double.
3. Given var pop:[String:Int], write a statement to add "Maryville", with a value of 11972, to pop.

4. What would the type of pop["Maryville"] be? Why is it optional?
5. What would the following output? (Assume that pop has several entries):

```
  for city in pop.keys {
         print(city, pop[city])
   }
```

6. How would you delete Maryville from pop?

Strings

1. What is the purpose of """ """ ?
2. How many characters are in this String? "\n\nHello!\\"
3. Write code to iterate through a String, **message**, and print the number of times a Character, **key**, appears in it. For instance, if message were "this is a test", and key was "t", you would print 3.
4. Wrap the code into a function, **count(**occurrencesOf:Character, within:Int). It should return the number of times key appears in message.
5. What is the value of "".isEmpty ?
6. Given message = "hello", what get's printed?

```
loc = message.startIndex
print( message[loc] )

loc = message.endIndex
print( message[loc] )

loc = message.index(before:message.endIndex)
print( message[loc] )

loc = message.index(message.startIndex, offsetBy:3)
print( message[loc] )

message.insert(contentsOf:"Oh,", at:message.startIndex)
print(message)

print( message.contains("llo") )
print( message.hasPrefix("Oh") )
print( message.hasSuffix("lo") )

message.lowercased()
```