**Question 1** - What have you learned recently about iOS development? How did you learn it? Has it changed your approach to building apps?

1. I have leanred about extensions to classes. I have also learned about filtering elemts in a collection.
2. By watching the keynote videos and exploring Stackoverflow. I also subscribe to several iOS dev weekly mailing lists to stay abreast of what is currently going on.
3. Extensions are a great idea when it becomes necessary to append to the underlying class of all objects. I am a little weary of extensions as I have yet to find a standardised best practice and there needs to be a line between subclassing and extending that I have yet to discover.

Filtering complex elements is incredibly usful when dealing with large data sets and reducing a collection to a specific set in a small amount of code. It makes the code easy to read and does not involve complicated for loops. I feel it makes the code much more readable for others who may have to maintain the code later. (e.g. collection.filter() {$0 > 2} rather than a for loop with an if statement).

**Question 2** - Can you talk about a framework that you've used recently (Apple or third-party)? What did you like/dislike about the framework?

I have used 2 frameworks recently: iCarousel and RSBarcodes\_Swift. What I like about iCarousel is the range of options that are available in terms of carousel styles. I find this better than Table or Collection views as it adds more to the user experience. What I am not impressed with is the fact that it only iterates images, and not entire view controllers. I think this is mainly due to the nature of animating the iteration of items to be quite intense on the system, so perhaps appending the library will have to wait until device can handle more.

RS\_Barcodes is a great library in that it produces barcodes in accordance with the standard AVFoundation classes found in Apple. I found that Apple was very versatile when reading a large range of types of barcodes, but lacking when generating the same types. This library uses all the same features of the built in code, making it very easy to use. I also like the fact that it uses CocoaPods, as this makes it easier to include in the app.

**Question 3** - Describe how you would construct a Twitter feed application [(here is an example of Udacity's Twitter feed)](https://www.udacity.com/api/nodes/5391950389/supplemental_media/exampletwitterfeedpng/download) that at minimum can display a company's Twitter page. Please include information about any classes/structs that you would use in the app. Which classes/structs would be the model(s), the controller(s), and the view(s)?

For the model, I would create the necessary Twitter objects (possibly have the timestamp, the Tweet and other data that I may need). In addition I would have the API tightly bound to objects, where the API can store and retrieve the object from a Web Call. While the API and Data Objects are tightly bound, they would not be in the same class.

I would define the controller as an object that can be used to initiate the API to retrieve and populate the objects, and return those model objects to the view (a ViewController, more specifically, a UITableViewController) to be displayed and retrieve user interaction.

The diagram below indicates a standard Model-View-Controller pattern, which I would employ for a simple application. There are other options, such as Model-View-Presenter and Model-View-ViewModel, which have certain properties that may make them a better choice in certain circumstances. All of these patterns are employed to make the app more maintainable.

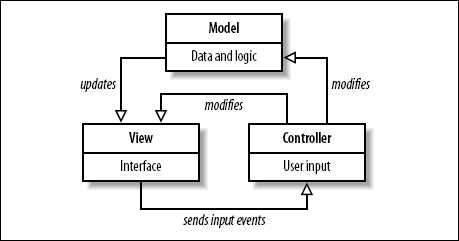


Figure 1: http://www.moock.org/lectures/mvc/images/mvc-02.jpg

**Question 4** - Describe some techniques that can be used to ensure that a UITableView containing manyUITableViewCell is displayed at 60 frames per second.

1. Keep the TableViewCells in a predefined style – using the standard cell style means that there is no custom layout happening on the cell where layouts need to be specifically measured
2. Keep cells simple – adding many items to a cell will impact the speed at which each cell can be rendered. Things like alpha values, additional subviews will impact the performance of the rendering.
3. Use a reuse identifier and retrieve cells off the “pool” of cells rather than creating a new cell every time the cellForRowAtIndexPath method is called – this enable the underlying system to reuse existing objects in memory rather than having to allocate additional memory at runtime.

**Question 5** - Imagine that you have been given a project that has this [ActorViewController](https://www.udacity.com/api/nodes/5391950389/supplemental_media/actorviewcontrollerswift/download). TheActorViewController should be used to display information about an actor. However, to send information to other ViewControllers, it uses NSUserDefaults. Does this make sense to you? How would you send information from one ViewController to another one?

This does not make sense to me. According to the Apple documentation on NSUserDefaults I believe this model violates the point of NSUserDefaults:  
As far as I am aware, when a ViewController wants to pass information from one ViewController to another, the programmer should override the **prepareForSegue** method. Inside this method, a reference to the next ViewController, or NavigationController, can be obtained from a destination object. The programmer can then set public *var* instances of the ViewController / NavigationController.

The NSUserDefaults class provides a programmatic interface for interacting with the defaults system. The defaults system allows an application to customize its behavior to match a user’s preferences.

There are three options for the manner in which the data can be passed:

1. Individual variables – passing the elementary data in single values between two View Controllers. I feel this would be a bad approach as it is not very maintainable and the data is likely to be grouped later anyway.
2. Passing a struct – the creating of a struct means that data and methods can be grouped together. The advantage of a struct is that it is passed by VALUE rather than by reference and will therefore result in a ecosystem with less chance of memory leaks. This is the method I would recommend for passing the data.
3. Passing an object – similar to a struct in terms of data and method grouping, but different in that data will be passed by reference, allowing multiple objects to maintain a reference to the object, which may cause memory leaks. However the advantage of the approach is that objects allow type checking at runtime and inheritance will allow complex object to assume properties and methods of another object.

**Question 6** - Imagine that you have been given a project that has this [GithubProjectViewController](https://www.udacity.com/api/nodes/5391950389/supplemental_media/githubprojectviewcontrollerswift/download). TheGithubProjectViewController should be used to display high-level information about a GitHub project. However, it’s also responsible for finding out if there’s network connectivity, connecting to GitHub, parsing the responses and persisting information to disk. It is also one of the biggest classes in the project.

How might you improve the design of this view controller?

I would separate the ViewController into a Model-View-Controller pattern. The API, Parser and model objects should be separated out.

I would create an GitHubAPI class, that can download and parse information, tightly coupled (yet independent in terms of file) to a model object. The API would check for connectivity and download the data, and use a generic to parse the information. The generic (or template) can remove the implemtation details of the parsing on to the specific object that requires parsing.

The GitHubProjectViewController is the View class, and should sent input events to the controller. The controller can respond to events by initiating the model. The model will download raw data through an API, send the data to a generic parser, which will end up creating platform agnostic data objects (Plain Old Swift Obejects). These objects will then be passed back to the Controller in the form of a result. The Controller can then decide which of these objects can be returned to the View and request the view to display them appropriately.

Data

Mod

Controller

View

Parser

API

GitHub

**Before answering the final question, insert a job description for an iOS developer position of your choice!**

Your answer for Question 7 should be targeted to the company/job-description you chose.

## Software Engineer, Mobile iOS

(Menlo Park, CA)

Facebook was built to help people connect and share, and over the last decade our tools have played a critical part in changing how people around the world communicate with one another. With over a billion people using the service and more than fifty offices around the globe, a career at Facebook offers countless ways to make an impact in a fast growing organization.

Every month, more than 680 million people around the world access Facebook through their mobile devices. As a leader in the mobile space, we constantly push the boundaries of what is possible in this fast-paced industry that is evolving daily. Facebook is seeking full-time iOS Engineers to join teams across the engineering organization to help take our iOS products to the next level. As an iOS Software Engineer, you will specialize in building elegant products on world-class technologies that bring the Facebook experience to hundreds of millions of people- anytime and anywhere.   
  
We are looking for self-starting engineers with strong experience developing sophisticated applications on the iPhone or iPad using the iOS SDK. If you are interested in joining a team of passionate, extremely talented industry veterans in the mobile space, who like to work hard and play hard, we look forward to hearing from you soon!

#### Responsibilities

* Work closely with our product and design teams to customize the Facebook experience for the iOS platform
* Implement custom native user interfaces using the latest iOS programming techniques
* Build reusable iOS software components for interfacing with the Facebook platform
* Analyze and optimize UI and back-end application code for efficiency and performance

#### Requirements

* B.S. or M.S. Computer Science or related field
* 3+ years of object-oriented software development experience
* 2+ years building complex applications for the iPhone or iPad using Objective-C/C++ with Cocoa and other frameworks
* 2+ years mobile application development at the user interface and system levels
* Ability to understand and debug large and complex code bases
* Experience designing clean and maintainable APIs
* Experience with multithreading programming
* Experience writing unit tests and testable code
* Knowledge of iOS SDK performance tools sand optimization techniques
* Excellent problem solving, critical thinking and communication skills

**Question 7** - If you were to start your iOS developer position today, what would be your goals a year from now?

1. To be an involved member of a core team

*I would like to be inserted into a team that regularly delivers content to the end user / product owner. The main goal is that something useful and satisfying is being incrementally created, updated or appended in a manner that provides use to subject of the delivery.*

1. To be improving myself through extra learning

*I intend to carry on with my learning, taking additional courses on both app architecture, software design patterns and good practice.*

1. To be on an architectural career path

*Through learning, hard work and mentorship I would make my career goals known to my peers and line manager so that they can encourage, teach and aid me in achieving my goals. I have found it very useful to have a meeting every three months with my line manager to make sure that I am staying on the path. I feel that the current position is the perfect place to start a career path that takes me in this direction. If I was placed in this position, I would be able to learn the current architecture of Facebook’s apps and learn what their current best practice is. I would also make it my goal to constantly refactor and strive to improve on the existing systems.*

1. Be involved in more than just simple development, but in the training of others within my space

*Through code reviews and daily work life, it is very important to me to be on good terms with my colleagues. Being a good developer is not always about writing code, but helping others to create good teams.*