Challenge 2: Time To Dig In

I hope everything went well with your last challenge! Be sure to send us your responses to the challenge questions if you haven't already!

Now you're going to start getting your hands dirty. Go ahead and find the lynda.com video series called: *Foundations of Programming: Object-Oriented Design*. Watch each video in the following list and answer to the correlated question. And again, send us your answers when you have completed them. As always, let us know if you get stuck!

Section 0 - Introduction

0.1 What to expect from this course 3m 6s
What is the intended purpose and potential advantage of learning object oriented design?

0.2 Exploring object-oriented analysis, design, and development 1m 41s
Why might it be advantageous to analyze and design before beginning programming?

0.3 Reviewing software development methodologies 4m 8s
What is the difference between a "waterfall" and an "agile" approach to development? What is an iteration and how do we to use them to build software?

Section 1 - Core Concepts

1.1 Why we use object-orientation 2m 42s
What are the various types of programming languages and in which domain is each used?

1.2 What is an object? 5m 22s
Describe in your own words the three properties of a computing object.

1.3 What is a class? 4m 43s
Explain how classes are analogous to blueprints. Include the relationship between a class and an object. Can you think of how the analogy breaks down?

1.4 What is abstraction? 2m 45s
When a developer uses the term "abstraction" what are they describing?

1.5 What is encapsulation? 3m 45s
What does encapsulation prevent? What does it enable?

1.6 What is inheritance? 3m 35s
Describe the inheritance relationship between classes. When would this relationship be advantageous to establish?

1.7 What is polymorphism? 3m 22s
What is the basic idea behind polymorphism? How can it make the classes we create more flexible?

Section 2 - Object-Oriented Analysis and Design

2.1 Understanding the object-oriented analysis and design processes 4m 13s
What are the steps of analysis that come before writing code for an application? Why do you think these steps make writing the code easier?

2.2 Defining requirements 6m 9s
What should you have after you've completed the first phase of defining your requirements?

2.3 Introduction to the Unified Modeling Language (UML) 1m 54s
What is UML? Why Is it useful to visualize your application before coding it?

Section 3 - Utilizing Use Cases

3.1 Understanding use cases 6m 11s
Write a use case for creating an event on your phone's calendar.

3.2 Identifying the actors 4m 16s
Can you think of a use case for a mobile application in which the actor is not the user of the mobile device?

3.3 Identifying the scenarios 5m 7s
Write another use case for a mobile device user interacting with a calendar application. This time include a couple extensions when crafting your scenario.

3.4 Diagramming use cases 4m 18s
Do a google image search for "use case diagram." Notice how many variations there are. What do they all generally have in common?

3.5 Employing user stories 3m 43s
Write 5 user stories to describe a mobile user interacting with his or her maps application.

Section 4 - Domain Modeling (Modeling the App)

4.1 Creating a conceptual model 1m 59s
Just let it soak in. No questions here.

4.2 Identifying the classes 2m 27s
Identify the classes in the use case you constructed for a user interacting with his or her calendar application in chapter 3.

4.3 Identifying class relationships 2m 38s
Identify the relationships among the classes you found above. Create a conceptual model where you diagram these relationships and then upload a picture of your model below.

4.4 Identifying class responsibilities 6m 43s
Identify the responsibilities of the classes you found above. List them here.

4.5 Using CRC cards 2m 49s
If you'd like, try creating CRC cards for the model you made above. There's no need to respond here, just try it out and see if you like this form of organization.

Section 5 - Creating Classes

5.1 Creating class diagrams 6m 11s
Construct Class Diagrams for the classes you imagine exist in a twitter app, a maps app, a calendar app, or any other app you would like to make. Do you find that it is easier to come up with the attributes or with the behaviors? Why do you think that is?

5.2 Converting class diagrams to code 4m 57s
How might the separation of interface and implementation in Objective-C be an advantage when working with class diagrams?

5.3 Exploring object lifetime 5m 55s
What are the constructors and destructors in Objective-C? Why do we use them?

5.4 Using static or shared members 5m 22s
Like the interest rate example in the video, give three additional examples of data that would be the same for all instances of a class.

Section 6 - Inheritance and Composition

6.1 Identifying inheritance situations 6m 49s

Describe in your own words what inheritance is and how it is useful when constructing classes.

6.2 Using inheritance 2m 43s

Referring to the apps on your phone, come up with three examples where you believe methods are being inherited from superclasses and called by subclasses.