Project 1: Soccer League Coordinator in Swift Study Guide

Sections of this Guide:

- **How to approach this project** includes detailed guidance to help you think about how to organize your code, project and files.
- **How to succeed at this project** lists the grading requirements for the project, with hints, links to course videos to refresh your memory and helpful resources.
- **GitHub resources**. Links to help you with uploading your project to GitHub.

How to Approach This project

Divide up the team players and assign to teams

Let's think about how we would do this in a real life situation rather than jumping into implementation right away. Basically we have 18 players, how would we assign them to three teams fairly by experience level? Here's one way to think about this:

- Divide the players into two groups, one for the experienced players, another one for the inexperienced

- Afterwards we can go through (iterate) each group alternately and assign players to the tree teams

How to succeed at this project

Here are the things you need to do pass this project. Make sure you complete them ***before*** you turn in your project.

- ❑ **Create required collections**
  First of all we have some modelling to do. Modelling means designing the structures in code to represent real world objects. So what do we have to model here? Looking at the approach above, we have:
  - ❑ Players
  - ❑ Experienced and Inexperienced groups
  - ❑ Teams
  - ❑ The League

Need help? Visit the Project 1 Slack Channel

Each player has a name, experience indicator, height, and guardian information. We can model each player with a Swift Dictionary. This is one way to do it:

```
let player1: [String : Any] = [ "name": "Joe Smith", "height": 42.0,
"experience": true, "guardians": "Jim and Jan Smith"]
```

The experienced group, inexperienced group, teams, and the league are all collections of players, hence they can all be modelled as an initially empty collection of Dictionaries, for example:

```
var experiencedPlayers: [[String: Any]] = []
...
var teamDragon: [[String: Any]] = []
...
var TheLeague: [[String: Any]] = []
```

- ❑ *Related video:* Swift Collections and Control Flow
- ❑ *Related resource:* Apple's The Swift Programming Language - Collection Types
- ❑ *Related resource:* Swift Collections

❑ **Create satisfactory logic**

Next is to implement the above mentioned approach to divide up the players and assign them fairly to teams. One way to do this is:

- ❑ Add all the players to The League collection
- ❑ Iterate through The League and check each player's experience level
  - ❑ If a player is experienced, add the player to the experienced collection
  - ❑ Similarly, if a player is inexperienced, add the player to the inexperienced collection
- ❑ After that, we can iterate through the experienced and inexperienced collection and assign players alternately to each of the 3 teams

- ❑ *Related video:* Swift Collections and Control Flow
- ❑ *Related resource:* Swift; How to iterate through all elements in an Array, Set, or Dictionary
- ❑ *Related resource:* How to sort an array using sort()

- **Personalized Letters**

  Lastly, we need to create and output the guardian letters. For the purpose of this project, you can simply iterate through each of the three teams, construct the content of the letters, and output them in the console.

  One note is that the content of the letters cannot have the word “Optional” in them. Please see the additional reference below for pointers on how to handle this properly.

  - *Related video:* Functions in Swift
  - *Related resource:* Swift Optionals

- **Naming and Comments**

  In order to clearly communicate to the persons who read your code, it is important to use clear and sensible variable names and also put in useful and meaningful comment throughout the code.

GitHub Resources

- GitHub and Xcode 9 integration
- Getting started with GitHub