By: Eng. Elias Al-Showaiter

[Dart Programming Language] - Lecture [14]

[Final Review]

1. After you finish this course you should have knowledge in these following topics:

   a. Variables: understand deeply what are variables, how to define them, what are data types, what values can I assign to them, what is the scope of a variable and how to follow a variable life cycle.

   b. Program Structure: you should understand how to structure your program and what each section means and when is it used and how is it used.

   c. Lists and Maps: you should be able to determine when do we need to use lists or maps. And have knowledge in multiple built in attributes and functions of these data types

   d. Functions: you should understand how to define functions and how to call them, understand how to determine the return type of the function, how to determine the parameters of the function.

   e. Classes: you should have knowledge in what are classes, when do I need them, what does a class consist from, what are the class attributes, constructor and methods. How to create objects from the class. What is inheritance, encapsulation, polymorphism.

f. Iterators: you should be familiar with functions that iterate over collections of data (lists, maps), like:

   i. forEach: iterate over the items and performs a task, but does not return a value

   ii. map: iterate over the items and performs a task and returns a value

   iii. where: iterate over the items and returns some of the items based on a condition

   iv. firstWhere: same as where but returns the first item that satisfies the condition

   v. reduce: iterates over the items and return a single items based on a condition

   vi. fold: iterates over the items and return a single value based on an operation.

g. Lambda Functions: you should be knowledgeable in how to use Lambda functions, what are its parameters, what do they return, when do I use the.

h. Null Safety: underhand how to create a nullable variable, and why should we define nullable variables.

i. Future Functions: understand when we use future functions and how to handle them, how to use async/await and how to use the then() function.

j. try/catch: understand the syntax and importance of try and catch.

k. GitHub: have a very limited understanding of GitHub, what does it do in general, and how to download files from it.

l. Follow the principles and guidelines of a good programmer and avoid bad practices.

m. Be able to apply these concepts across any other programming language.

n. Develop skills for problem solving

2. What is expected from you in the future, when we study flutter:

   a. Development Environment: you should have flutter installed and ready to use.

   b. Code Analysis: you should be able to look at a block of code and have some basic insight on what does this code does.

      i. Determining each variable data type

      ii. Determining what each function performs

      iii. What are the attributes and methods of the classes

   c. Understanding programming Terms: you should be able to follow with code when it is explained to you using Programming terms for example, parameters, arguments, attributes, methods, constructor, data types, objects.

d. Basic Look at the Future: knowing what we can do with classes, lists and iterators, you should have a slight understanding how complex applications handle data.
e. Depending on yourself: you should have confident and trust yourself in taking any challenge.

f. Continue Building your programming skills on a firm ground.

3. What is coming Next:

a. Frontend Programming: now that you have strong understanding of the basics it is the perfect moment to start taking a look at how design screens and how to move your skills into the next step.

b. A new Style: designing screens in the frontend is a very different experience, it uses different style so it might be a little challenging in the beginning, however once you become familiar to it, it becomes more enjoyable.

c. New Framework: in flutter we will have a new set of functions, classes. Almost everything is built for us and we just need to use them in the appropriate way

d. New Challenges: when designing frontend usually the hardest challenge is not avoiding errors or understanding the code. It is to make our vision come true using the code.

e. GitHub: having more experience with Git & GitHub, you will be uploading your assignments into GitHub yourself.