**Chu Anh Lợi – 20215280**

*Lesson 2*

[1. Explore the main() function](https://developer.android.com/codelabs/android-development-kotlin-2.1#0)

Step 1: Create a Kotlin file

A screenshot of a computer program

Description automatically generated

Step 2: Add code and run your programA screenshot of a computer

Description automatically generated

Step 3: Pass arguments to main()

A screenshot of a computer

Description automatically generated

Step 4: Change the code to use a string template

A screenshot of a computer

Description automatically generated

[2. Learn why (almost) everything has a value](https://developer.android.com/codelabs/android-development-kotlin-2.1#1)  
A screenshot of a computer program

Description automatically generated

[3. Learn more about functions](https://developer.android.com/codelabs/android-development-kotlin-2.1#2)

Step 1: Create some functions

A screenshot of a computer

Description automatically generated

Step 2: Use a when expression

A screenshot of a computer program

Description automatically generated

[4. Explore default values and compact functions](https://developer.android.com/codelabs/android-development-kotlin-2.1#3)

Step 1: Create a default value for a parameter

A screenshot of a computer program

Description automatically generated

Step 2: Add required parameters

A screenshot of a computer program

Description automatically generated

Step 3: Make compact functions

A screenshot of a computer program

Description automatically generated

Nhận xét: Output từ println() với shouldChangeWater() giống như trước khi chuyển sang sử dụng các hàm compact.

[5. Get started with filters](https://developer.android.com/codelabs/android-development-kotlin-2.1#4)

Step 1: Create a filter

A screenshot of a computer program

Description automatically generated

Step 2: Compare eager and lazy filters

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

[6. Get started with lambdas and higher-order functions](https://developer.android.com/codelabs/android-development-kotlin-2.1#5)

Step 1: Learn about lambdas

A screenshot of a computer program

Description automatically generated

Step 2: Create a higher-order function

A screenshot of a computer program

Description automatically generated

A screen shot of a computer program

Description automatically generated

A screen shot of a computer

Description automatically generated

[7. Summary](https://developer.android.com/codelabs/android-development-kotlin-2.1#6)

* Để tạo các tệp nguồn Kotlin mới trong IntelliJ IDEA, nhấp vào "src" trong bảng Project và nhấp chuột phải để mở menu. Chọn "New -> Kotlin File/Class".
* Để biên dịch và chạy một chương trình trong IntelliJ IDEA, nhấp vào tam giác màu xanh lá bên cạnh hàm `main()`. Kết quả sẽ xuất hiện trong một cửa sổ bên dưới.
* Trong IntelliJ IDEA, bạn có thể chỉ định các đối số dòng lệnh để truyền vào hàm `main()` trong Run > Edit Configurations.
* Hầu như mọi thứ trong Kotlin đều có giá trị. Bạn có thể sử dụng điều này để làm cho mã của mình ngắn gọn hơn bằng cách sử dụng giá trị của một câu lệnh `if` hoặc `when` như một biểu thức hoặc giá trị trả về.
* Các đối số mặc định giúp loại bỏ nhu cầu tạo nhiều phiên bản của một hàm hoặc phương thức. Ví dụ: `fun swim(speed: String = "fast") { ... }`
* Các hàm ngắn gọn, hoặc hàm đơn biểu thức, có thể làm cho mã của bạn dễ đọc hơn. Ví dụ: `fun isTooHot(temperature: Int) = temperature > 30`
* Bạn đã học một số điều cơ bản về bộ lọc, sử dụng biểu thức lambda. Ví dụ: `val beginsWithP = decorations.filter { it[0] == 'p' }`
* Một biểu thức lambda là một hàm không bị ràng buộc với một định danh, tức là một hàm ẩn danh. Biểu thức lambda được định nghĩa trong dấu ngoặc nhọn {}.
* Trong một hàm bậc cao, bạn truyền một hàm, chẳng hạn như biểu thức lambda, vào một hàm khác dưới dạng dữ liệu. Ví dụ: `dirtyLevel = updateDirty(dirtyLevel) { dirtyLevel -> dirtyLevel + 23 }`

\_\_HẾT\_\_