Programming in Kotlin

Kotlin is the now the standard programming language for Android. It is very similar to Swift, which is used in iOS programming. In fact, if you learn Kotlin, then you should be able to write a program that is almost the same in Swift.

Variable declarations

Kotlin uses type inferencing, meaning that you declare a variable using the ***var*** keyword and then make it equal to a value, like:

```
var i = 0
var j = "Hello world"
```

In this example, i is an **int** because what comes after the = is 0, which is an int. The variable j is a String because what comes after

You can also assign a type to a variable like this:

```
var i : Integer = 0
var j : String = ""
```

The format is:

```
var name : Type =  initialValue
```

Notice as well that Kotlin doesn't require semi-colons ; at the end of a line of code.

To make a variable a constant which can't be changed, use **val** instead of **var**.

```
val i  = 0
val j = ""
```

Function declarations

Kotlin uses the **fun** keyword to declare a function:

```
fun aFunction()
```

The return type of a function is similar to variable declarations, where the return type comes after the name:

```
fun myFunction() : returnType {   }
```

If a function has no return type, then you don't have to declare it.

Function parameters are declared as in UML, where the name comes first, and then followed by the variable type:

```
fun myFunction(param1:String, param2:Int, param3:String) {   }
```

Let's practice now by writing a function called "printStrings", which takes two string parameters: s1, and s2:

```
fun printStrings(s1:String, s2:String) {   }
```

Kotlin makes it easy to embed strings inside of other strings. Instead of Java's String concatenate, Kotlin uses the $ followed by the variable that you want to put in a string. For instance, our example function of printStrings, we can use **"String 1: $s1 String 2:s2"**