## CET3013N: - Coroutines

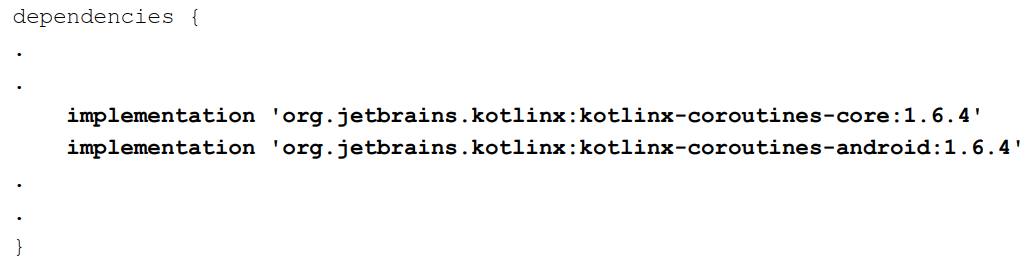
**Week 7 Exercises: Coroutines**

###### Objectives – What this lesson is trying to achieve.

Creating the Coroutine Application

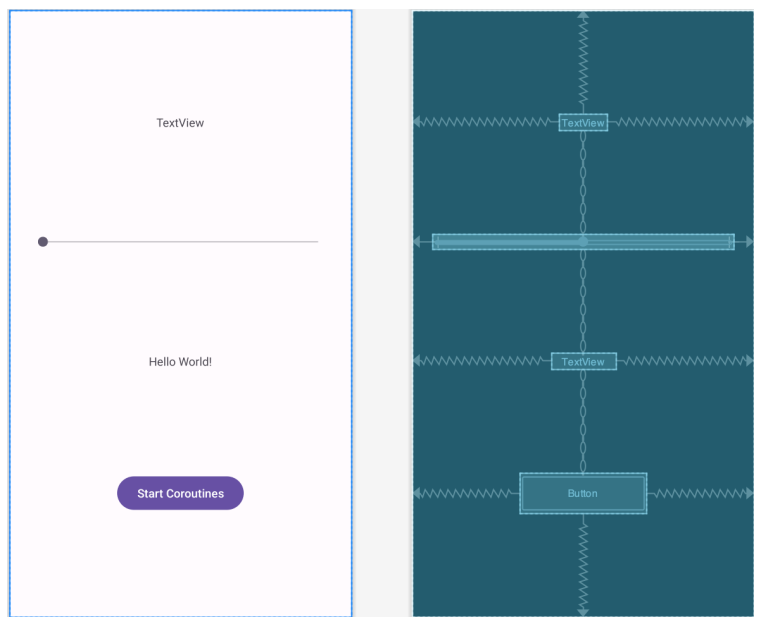
### Task 1. Creating Project

1. Create a new empty project and call it **Week 7** **Coroutines**. You can leave all the settings at their defaults as usual.
2. Migrate the project to view binding.
3. Include support for coroutines in newly created projects. Edit the Gradle Scripts -> build.gradle (Module :app) file and add the following lines to the dependencies section. Please check the current version of library file.



### Task 2. Designing User Interface

1. Begin by loading the activity\_main.xml layout file and add the Button, TextView, and SeekBar objects so that the layout resembles that shown as below:



1. Set the appropriate id for each GUI widget.

|  |  |
| --- | --- |
| **GUI Widgets** | **id** |
| First TextView | countText |
| SeekBar | seekBar |
| Second TextView | statusText |
| Button | startButton |

1. The SeekBar controls the number of asynchronous coroutines, ranging from 1 to 2000.

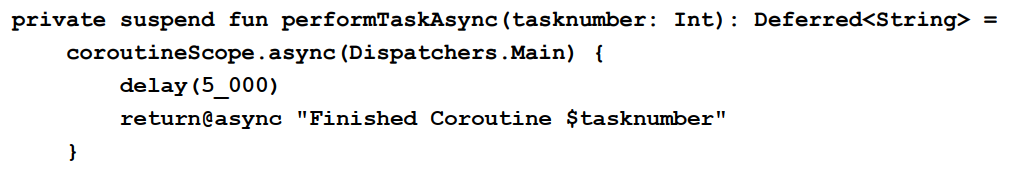
### Task 3. Coroutines Implementation in MainActivity

1. Set the view binding for the layout.
2. Declare an integer variable called **count** with the initial value **1**.
3. Implement the **SeekBarChangeListener** for the SeekBar and override all three methods (**onProgressChaged**, **onStartTrackingTouch** and **onStopTrackingTouch**.

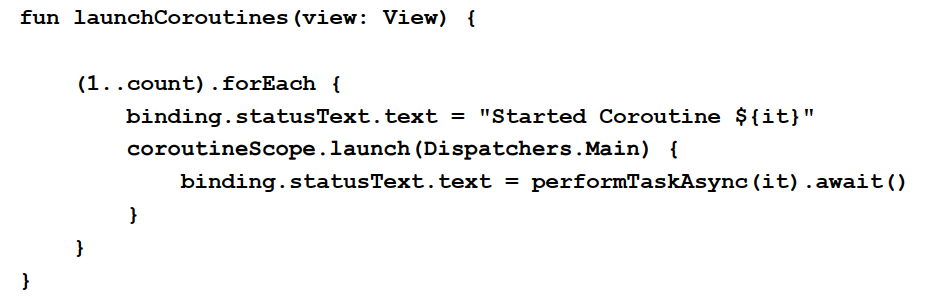
1. In the onProgressChanged method, update the countText with the no of corountines that will be executed.
2. Update the **count** variable with the **progress** value.
3. Create a customised coroutine scope in the MainActivity.



1. Create a suspend function to delay for 5 seconds and then return a string indicating that the numbered couroutines has finished.



1. Add the **launchCoroutines**() method which is called when the Button object is clicked.



The method implements a loop to launch the requested number of coroutines and updates the status TextView each time a result is returned from a completed coroutine via an await() method call.

1. Build and run the app on a device or emulator and move the SeekBar to a low number (for example 10) before tapping the launch button.

1. Repeat the process with the SeekBar set to 2000, this time sliding the seekbar back and forth as the coroutines run to verify that the main thread is still running and has not been blocked.