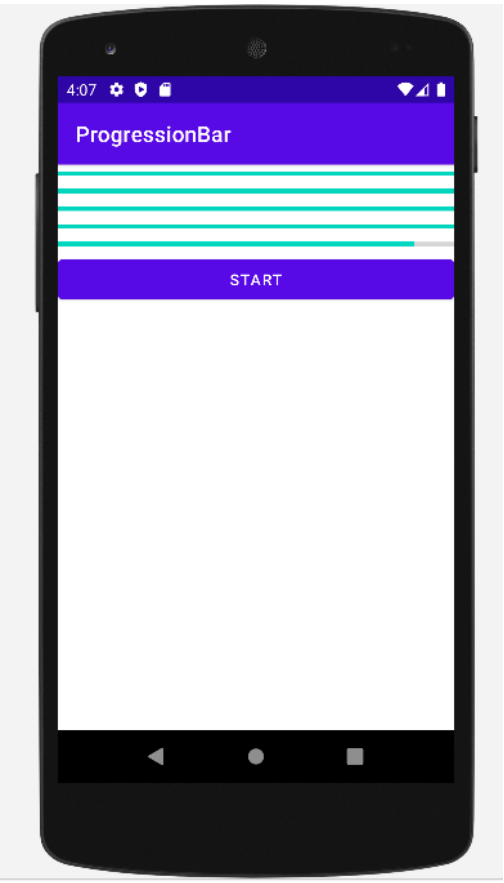
Lab 8: Threads, Handle and Looper

Since computers have multiple cores, we need to use threads to use all its potential. Java has implemented classes that allow us to run threads and create concurrency in our programs. When it comes to Android applications, we will always have the main thread that can also open other threads to run background operations.

Overloading the main thread can slow down the system and since it is responsible to update the User Interface when it is blocked the application could crash. The other important rule you should now is that you should not modify the Views classes (buttons, imageViews, editTexts, etc) from any thread except the main thread.



In this assignment we are going to use threads using two methods:
1.- Executor services: Which is a class provided by Java that simplifies the management of threads.
2.- Combining handlers and loopers: This combination is a bit more complex than the previous one, but updating the UI is the responsibility of Android OS

Part I: Executor service (1 point)

Executor service is a method you can use to run tasks in the background.
Download the code you have in Brightspace (ProgressionBar) modify it to do the following

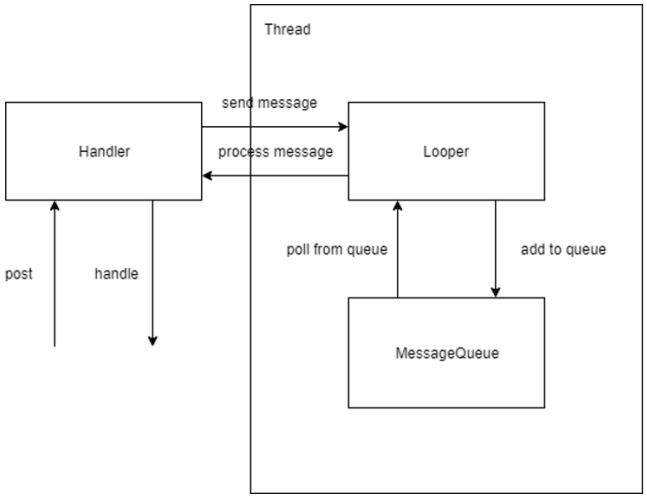
1. Add comments to the app indicating the most important functions and what they do. (0.3 pts)
2. How to change the number of threads that are executed at the same time? (**0.2 pts)**
3. Add a button to reset the progress of all the progress bar that had been completed.
   Also show in an Edit Text the percentage progress that has been done. For example, if 1 has been completed the progress is 20%, if two bars are completed show 40%, and so on. **(0.3 pts)**
4. What is happening when you click start again before it finishes? Can you think of a simple way to prevent this? (0.2 pts)

Part II: Handlers, Loopers and Threads (1 point)

The other way in which you can implement threads in Android OS is by using handlers, loopers, and threads.

Just a quick reminder of their functionality:

1. Looper: It is an object responsible for constantly moving along the “MessageQueue” to see if there are new messages. It uses the methods: prepare and loop.
2. Handler: It is an object that interacts with the looper and puts and removes messages in the queue of events.
3. MessageQueue: It is the queue with all the messages.



If you go to Brightspace you will find a file that already implements tasks (multiTasks.txt). The program is very simple, it uses a handler and a looper to execute tasks in the background using a thread.

Use this code to implement the previous example **without the modifications.** In other words, implement the program that runs the 5 progress bars using only the start buttons but instead of using ExecutorService, use the option of handlers, loopers, and threads.

References:

https://medium.com/@dcostalloyd90/handler-looper-thread-2463b11d3d44
https://android-coding.blogspot.com/2014/08/android-example-to-execute-threads-with.html