ITIS/ITCS 5180 Mobile Application Development
In Class Assignment 4

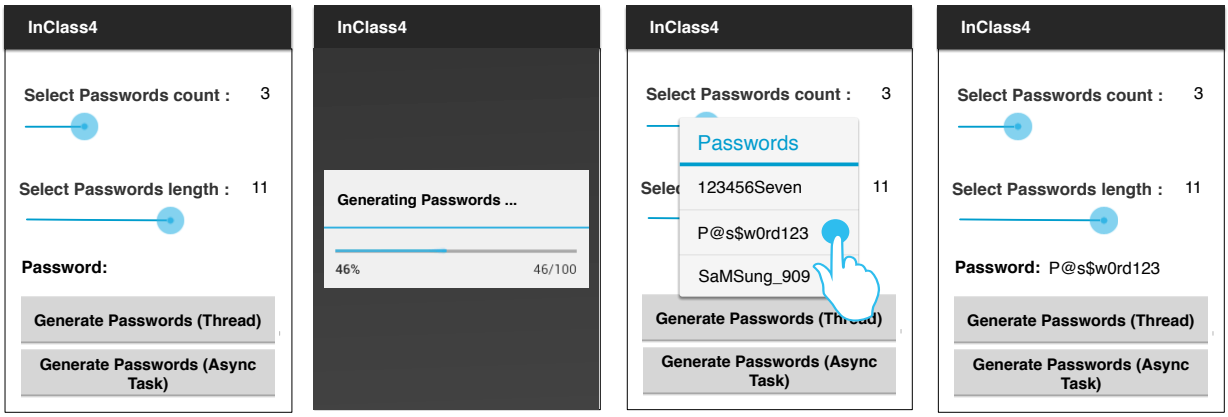
Basic Instructions:

---

1. In every file submitted you MUST place the following comments:
   a. Assignment #.
   b. File Name.
   c. Full name of all students in your group.
2. Each group should submit only one assignment. Only the group leader is supposed to submit the assignment on behalf of all the other group members.
3. Your assignment will be graded for functional requirements and efficiency of your submitted solution. You will loose points if your code is not efficient, does unnecessary processing or blocks the UI thread.
4. Export your Android project and create a zip file which includes all the project folder and any required libraries.
5. Submission details:
   a. Only a single group member is required to submit on canvas for each group.
   b. The file name is very important and should follow the following format: **Group#_InClass04.zip**
   c. You should submit the assignment through Canvas: Submit the zip file.
6. **Failure to follow the above instructions will result in point deductions.**

In Class Assignment 4 (100 Points)

In this assignment you will get familiar with Android concurrency models. The app is a password generator app to help the user to choose strong passwords. The User selects how many passwords they want the app to generate, the length of each password and then chooses one of them. This application is composed of a single activity, namely the Main Activity.



(a) Main Screen (b) Progress Dialog (c) Passwords List (d) Password selected

Figure 1, App Main Screen

Using Threads and Handlers (70 points)

The interface should be created to match the user interface (UI) presented in Figure 1. You will be using layout files, and strings.xml to create the user interface. Perform the following tasks:

1. You are provided with a Util class that contains a static method getPassword(int length). The length is the length of the password to be generated. This method is expected to take long time to execute and returns a random String password. Import the provided Java file by simply dragging the file into the src folder under your project package in Android Studio.
2. You should create a thread pool of size 2, and use the pool to execute your created threads.
3. There are two SeekBars in this assignment:
   a. Number of passwords SeekBar, this is used to set the number of passwords to be generated. The SeekBar minimum should be 1 and maximum 10. The TextView beside the seeker should show the selected SeekBar progress whenever the user changes the SeekBar progress (moves the SeekBar).
   b. Password length, this Seek bar is used to set the password length. The SeekBar minimum should be 8 and maximum 23. The TextView beside the seeker should show the selected SeekBar progress whenever the user changes the SeekBar progress (moves the SeekBar).
4. Tapping the “Generate Passwords (Thread)” button should start the execution of a background thread and return the list of all passwords (based on the selected count

and length) by using the getPassword() method. For example, if the count was set to 5, the getPassword() method will run 5 times in the background thread, and return 5 passwords. The list of these 5 passwords should be returned to the main thread and displayed in the AlertDialog. While the passwords are being generated, display a ProgressBar indicating the progress, see Figure1(b).
5. To be able to exchange messages between the child thread and the main thread use the Handler class. Either use messaging or setup a runnable message.
6. The ProgressBar should not be cancelable. The ProgressBar should be dismissed after all the getPassword() calls are completed. The list of passwords should be displayed using an alert dialog as shown in Figure 1(c).
7. Tapping one of the passwords generated should dismiss the Dialog and display the selected password as shown in Figure 1(d).

Part 2 (30 Points): Using AsyncTask

This part is similar to Part 1, but you should use AsyncTask to implement the same functionality provided by Part 1. Perform the following tasks:

1. Tapping the “Generate Passwords (Async Task)” button should start the execution similar to Part 1 but using Async Tasks.

Notes:

1. If you are using API level 26 or higher, the ProgressDialog has been deprecated. In this case use ProgressBar. https://developer.android.com/reference/android/widget/ProgressBar.html