ITIS/ITCS 4180/5180 Mobile Application Development
In Class Assignment 4
Spring 2016

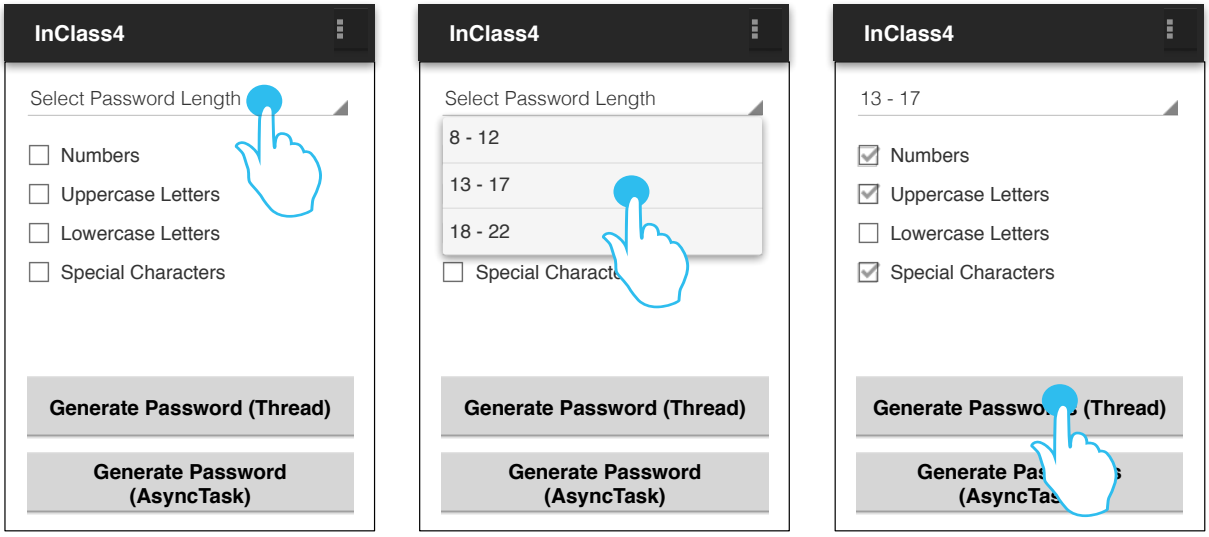
Basic Instructions:

---

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. Please download the support files provided with this assignment and use them when implementing your project.
5. Export your Android project and create a zip file which includes all the project folder and any required libraries.
6. Submission details:
   a. Only a single group member is required to submit on moodle 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 Moodle: Submit the zip file.
7. **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 the range for length of password they want the app to generate, chooses the type of characters they want in their passwords and then generates them. This application is composed of a single activity.



(a) Main Screen

(b) Selecting Password Length from spinner

(c) Selecting type of characters & generating password

(d) ProgressDialog

(e) Generated Password

Figure 1, App Main Screen

Part 1 (60 Points): Using AsyncTasks

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. Create a new android project called “In Class 4”.
2. You are provided with a Util class file that contains a static method getPassword(). This method takes a 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.
3. Your task is to use an AsyncTask to execute this method in a background thread. Do not use the main thread to generate passwords. The UI should be manipulated by the only main thread.
4. Use a Spinner to set the length of password. The length should be predefined and selected among these three values: 8-12, 13-17 and 18-22. This will decide the random length of the generated password between the given range. This selected spinner position value should be sent to the getPassword() function as a the “length” when the user clicks on the button: Selecting 8-12 should pass a 0 to getPassword(),13-17 should pass a 1, and 18-22 should pass a 2.
5. There should be checkboxes for Numbers, Uppercase Letters, Lowercase Letters and Special Characters. The selected values should be used to send to the getPassword() function as booleans. There MUST be at least one checkbox selected to execute the getPassword() function.
6. Tapping on the “Generate Password (AsyncTask)” button should start the execution of a background AsyncTask and generate the password according to provided values of Password Length and the selected checkboxes and then displays it on the screen in big red letters as shown in Figure1(e). For example, if the password length is set to 13-17 and the checkboxes of Numbers, Uppercase Letters and Special Characters are selected, the getPassword() method will return a password similar to that shown in the figure. While the passwords are being generated, display a ProgressDialog indicating the progress, see Figure1(d). The Async input parameter should be set to Integer, the progress parameter to Integer, and the result parameter to String.
7. The ProgressDialog should not be cancelable. The ProgressDialog should be dismissed after all the getPassword() calls are completed (in onPostExecute). The password should be displayed in TextView in large font with red color as shown in Figure 1(e).

Part 2 (40 Points): Using Threads and Handlers

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

1. Tapping on the “Generate Password (Thread)” button should start the execution of a background thread and return password (based on the input provided) by using the getPassword() method, as done in Part 1.
2. 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.