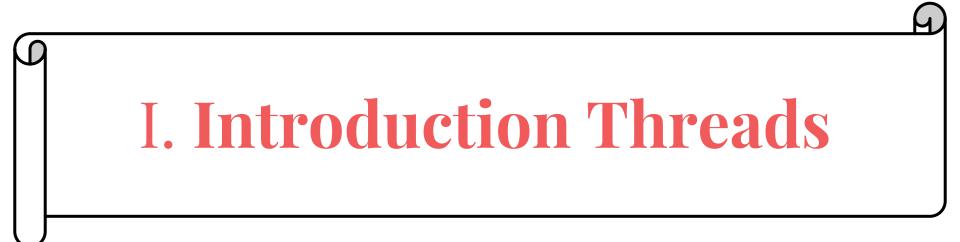
Android Advance
Lesson 4
Thread,
AsyncTask,
Handler



Outline

- I. Introduction Threads
- II. Execute asynchronous task
- III. AsyncTask
- IV. Handler





1. What is thread?

- > A thread is a thread of execution in a program. The Java Virtual Machine allows an application to have multiple threads of execution running concurrently.
- Every thread has a priority. Threads with higher priority are executed in preference to threads with lower priority. Each thread may or may not also be marked as a daemon. When code running in some thread creates a new Thread object, the new thread has its priority initially set equal to the priority of the creating thread, and is a daemon thread if and only if the creating thread is a daemon



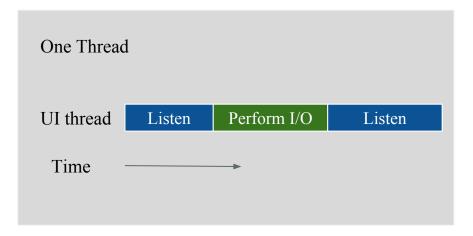
2. How thread work?

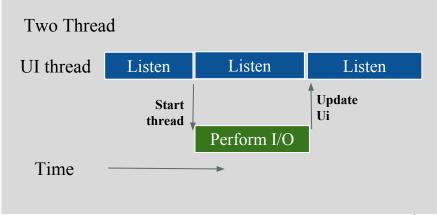
- Is a single sequential flow of control within a program.
- Often completes a specific task



Asynchronous task

- > By default an Android app use a single thread call the *UI thread* do display user interface (only UI thread can update UI)
- ➤ In Development process we need run heavy task, if run in UI thread UX is very bad. So we need run it in different thread





•





1. Execute asynchronous task

- Is an easy way to perform a background task and publish result on UI thread without having to manually manipulate threads
- Use generics to allow a class to operate on various type of objects.
- Provide for three generic type
 - Parameters
 - Progress
 - Result



2. Execute asynchronous task

Method	Is executed
onPreExcute()	On the UI thread before the task is execute
doInBackground(Params)	On the background thread immediately after the onPreExecute method finishes. Override this method to perform a computation on a background thread. The specified parameters are the parameters passed to execute(Params) by the caller of this task. This method can call publishProgress(Progress) to publish updates on the UI thread.
onProgressUpdate(Progress)	Runs on the UI thread after publishProgress(Progress) is invoked. The specified values are the values passed to publishProgress(Progress).
onPostExcute(Result)	Runs on the UI thread after doInBackground(Params). The specified result is the value returned by doInBackground(Params).





1. What is Handler?

- ➤ A Handler allows you to send and process <u>Message</u> and Runnable objects associated with a thread's <u>MessageQueue</u>
- There are two main uses for a Handler:
 - To schedule messages and runnables to be executed as some point in the future
 - To enqueue an action to be performed on a different thread than your own.

2. How to use Handler

```
Handler mHandler = new Handler() {
    @Override
    public void
handleMessage (Message msg) {
        if(msq.what == 0) {
            updateUI();
         }else{
            showErrorDialog();
```

```
Thread thread = new Thread() {
    @Override
    public void run(){
        doSomeWork();
        if (succeed) {
            //we can't update the
UI from here so we'll signal our
handler and it will do it for us.
    mHandler.sendEmptyMessage(0);
        }else{
            h.sendEmptyMessage(1);
```



Q&A

Exercise

- 1. Apply knowledge above do splash screen for Android App (ref: Youtube App For Android)
- 2. Continue Exercise in Android-Advance-Lesson-3. Require load image from library using AsyncTask