

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

**Basic Instructions:**

---

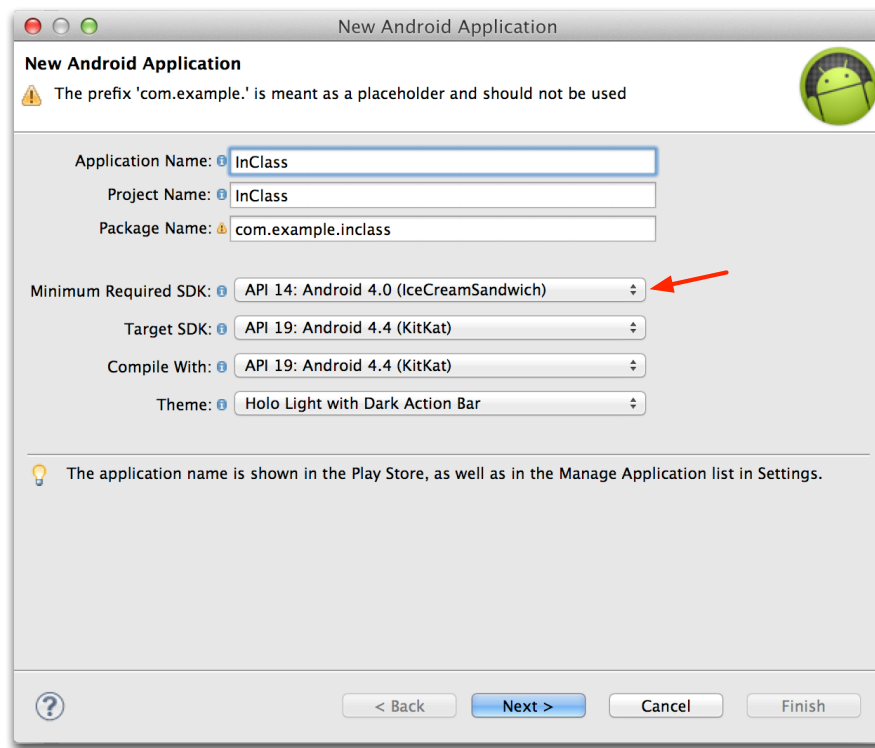
1. In every file submitted you **MUST** place the following comments:
  - a. Assignment #.
  - b. File Name.
  - c. Full name of all students in your group.
2. Each group should submit only one assignment. Only the group leader is supposed to submit the assignment on behalf of all the other group members.
3. Please download the support files provided with this assignment and use them when implementing your project.
4. Export your Android project as follows:
  - a. From eclipse, choose "*Export...*" from the File menu.
  - b. From the Export window, choose *General* then *File System*. Click *Next*.
  - c. Make sure that your Android project for this assignment is selected. Make sure that all of its subfolders are also selected.
  - d. Choose the location you want to save the exported project directory to. For example, your *Desktop* or *Documents* folder.
  - e. When exporting make sure you select *Create directory structure for files*.
  - f. Click Finish, and then go to the directory you exported the project to. Make sure the exported directory contains all necessary files, such as the .java and resource files.
5. Submission details:
  - a. When you submit the assignment, compress your exported Android project into a single zip file. The format of compressed file name is InClassAssignment#.zip
  - b. You should submit the assignment through Moodle: 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 build your first Android application. You will get familiar with simple multithreading and UI manipulation.

### Important App Requirements:

1. The required Android Virtual Device (AVD) should have **minimum SDK version set to 14 and target SDK at least 17**. The app should display correctly on 3.2" QVGA (ADP2) (320x480: mdpi). Your assignment will not be graded if it does not meet these requirements, and you will not be granted any points on your submission.



**Fig 1. Choosing Minimum Required SDK to 14**

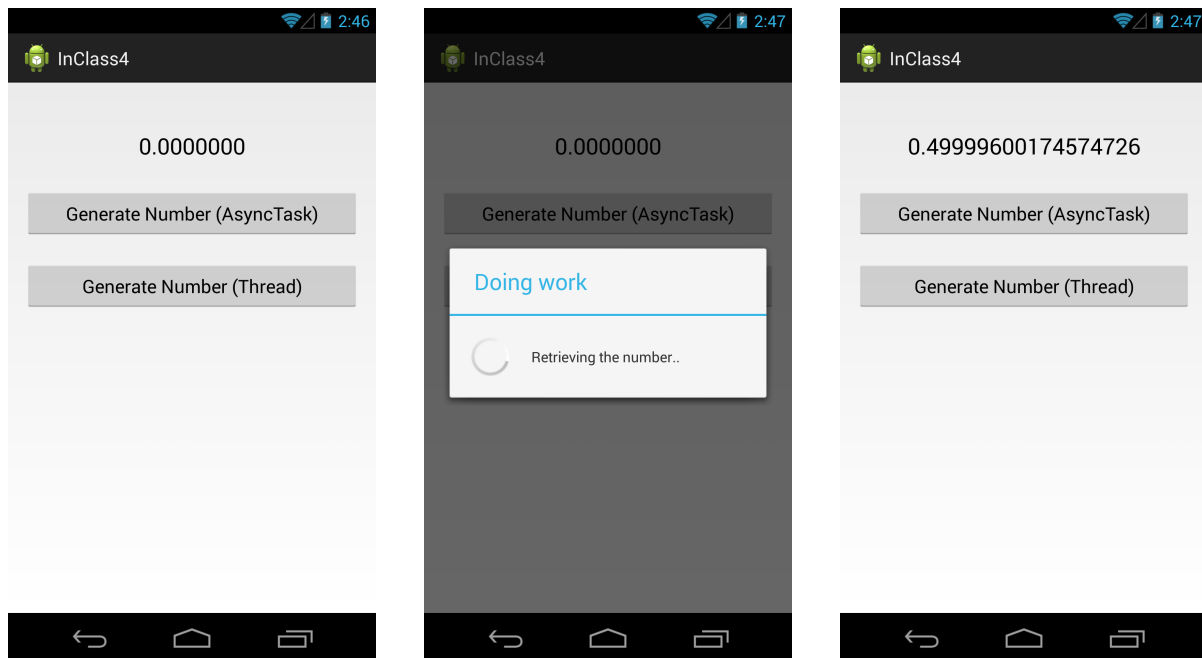
### Part 1 (50 Points): Using AsyncTasks

The interface should be created to match the user interface (UI) presented in Figure 2. 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 HeavyWork class that contains a method `getNumber()`. This method takes a long time to execute. Your task is to use an `AsyncTask` to execute this method in a background thread. The method returns a number which should be displayed in the `TextView` in the layout. Do not use the main thread to generate the number. The UI should be manipulated by the only main thread.
3. Tapping on the "Generate Number (AsyncTask)" button should start the execution of a background `AsyncTask` the computes the number using the `getNumber()` method in the `HeavyWork` class. While the number is being generated display a `ProgressDialog`

indicating that the number is being generated, see Figure2(b).

4. The ProgressDialog should not be cancelable. The ProgressDialog should be dismissed after the number is generated and displayed in the TextView, see Figure 2(c).



(a) Launching screen

(b) ProgressDialog

(c) Generated Number Updated

**Figure 2, Application User Interface (Part 1)**

## **Part 2 (50 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. Do not use the main thread to generate the number. The UI should be manipulated by the only main thread.
2. Tapping on the “Generate Number (Thread)” button should start the execution of a background thread the computes the number using the `getNumber()` method in the `HeavyWork` class. While the number is being generated display a ProgressDialog indicating that the number is being generated, see Figure1(b).
3. The ProgressDialog should not be cancelable. The ProgressDialog should be dismissed after the number is generated and displayed in the TextView, see Figure 1(c).
4. 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.