

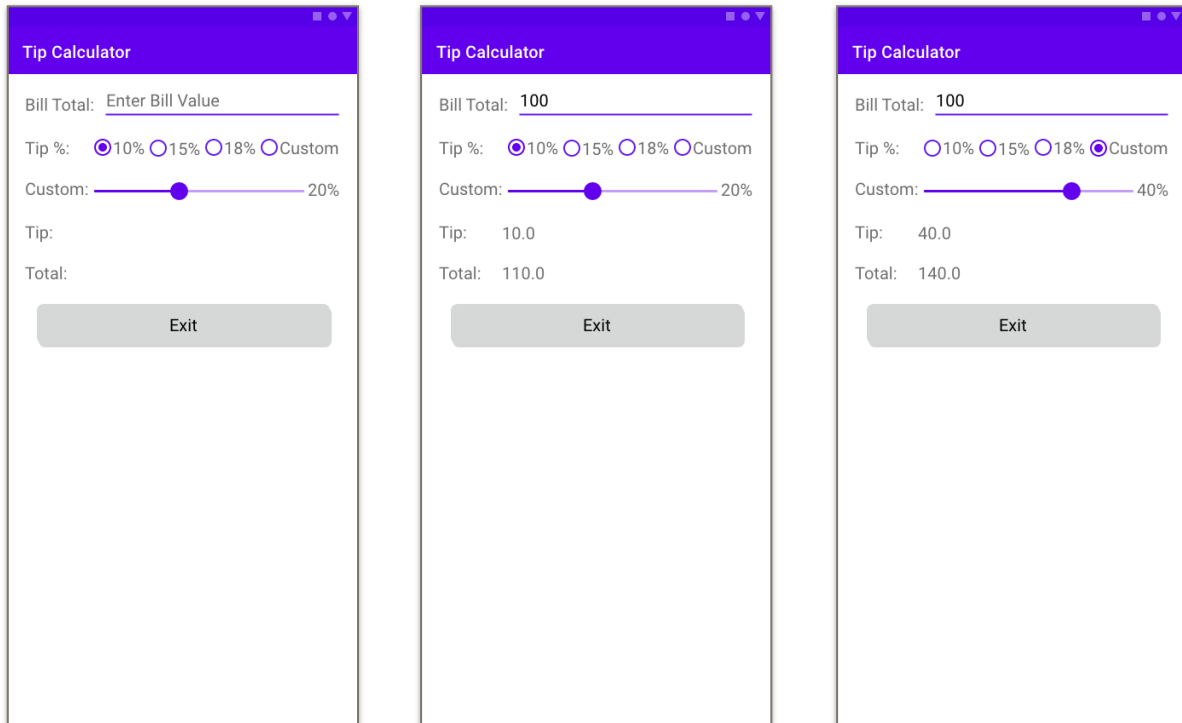
Mobile Application Development
Homework 1

Basic Instructions:

1. In every file submitted you **MUST** place the following comments:
 - a. Assignment #.
 - b. File Name.
 - c. Full name of the student.
2. Each group is required to submit the assignment on Canvas.
3. **Submit Codes:**
 - a. Zip all the project folder to be submitted on canvas.
4. Submission details:
 - a. The file name is very important and should follow the following format:
HW01.zip
 - b. You should submit the assignment through Canvas: Submit the zip file.
5. **Failure to follow the above instructions will result in point deductions.**

Homework 01 (100 points)

In this assignment you will build an Android application. You will get familiar with common Android components and how to interact with them. You will build a single activity Tip Calculator application.



(a) Initial App Screen

(b) Tip calculated using the 10% option

(c) Tip calculated using the custom option

Figure 1, Application Screen Shots

Part 1 (30 Points): Building the Interface

The interface should be created to match the user interface (UI) presented in Figure 1. You will be using layout files, strings.xml, and drawable files to create the user interface. The layout XML file can be modified through the raw xml, or through the GUI tools provided within eclipse. To build the UI, please follow the following tasks:

1. Your application should have an application launcher icon, please select your launcher icon to represent your app.
2. The string values used for the text labels, button labels and hints should be read from the strings.xml file and should not be hardwired in the layout file.
3. Use an EditText component for the user to enter the bill total value. The EditText component should be setup to limit the bill total value to only positive numbers. When the app starts the bill total EditText should be empty, and should display the hint message "Enter Bill Value" as indicated in Figure 1(a).
4. Use a RadioGroup containing RadioButtons to enable the user to pick from the tip options 10%, 15%, 18% and Custom. When the app starts the 10% tip choice should

be selected.

5. Use the SeekBar to enable the user to pick a custom tip value, the maximum custom tip value should be set to 50%. When the application starts the custom tip value should be set to 20%. On the right of the SeekBar use a TextView to display the current progress of the SeekBar, which represents the current custom tip value.
6. Use TextView components for creating the tip and total values. When the app starts the tip and total values should be both set to empty strings as shown in Fig 1(a).
7. Finally, create the Exit button which should finish the activity if pressed.

Part 2 (70 Points): Event Handlers and App Behavior (MainActivity)

In this part you will build the required logic for the tip calculator app. The requirements are as follows:

1. The tip radio buttons should enable the user to select one of tip options. If a tip option is selected the tip and total values should be calculated and update based on the selected tip option and the bill total. See Figure 1.
2. If the user updates or edits the bill total the tip and total values should be updated to reflect their new values.
3. If the bill total is empty, then tip and total values should be set to empty strings. In addition, show a toast message displaying an error message informing the user to "Enter Bill Total".
4. When the custom SeekBar is changed the current custom tip progress should be updated to show the SeekBar progress. If the custom tip option is selected, then the tip and total values should be calculated based on the current custom tip progress.
5. The application should be finished if the Exit button is clicked.