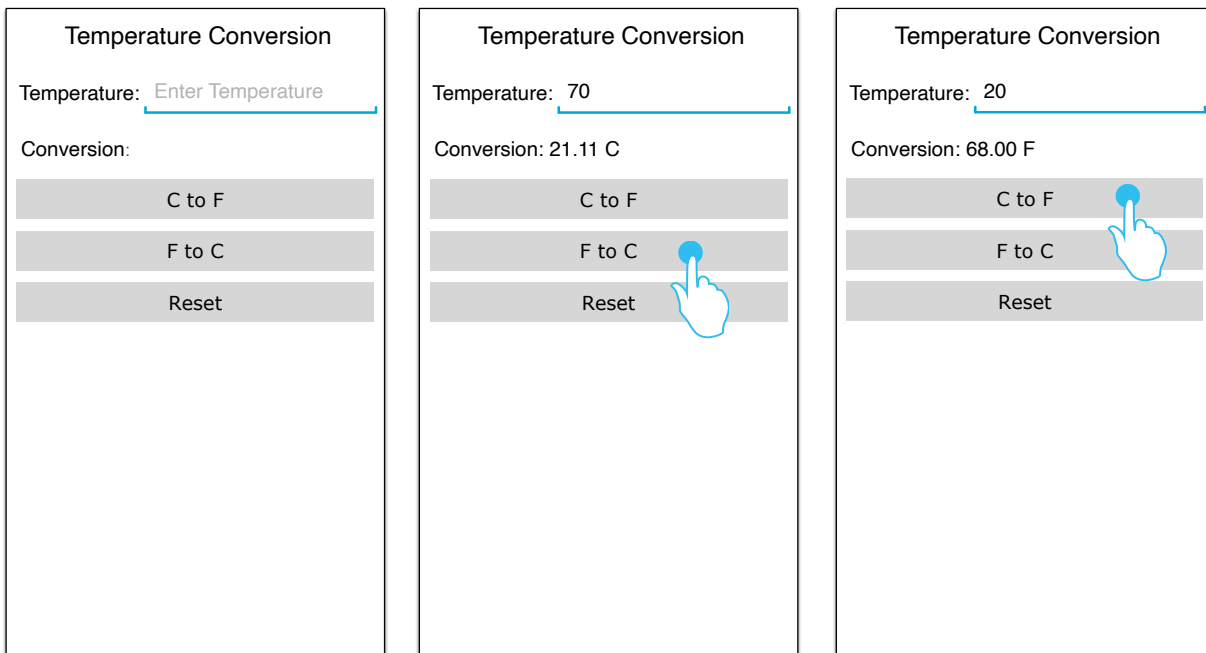


Mobile Application Development
Assignment 1 (100 Points)

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:
Assignment#.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.**

In this assignment you will build a simple mobile app that enables the temperature conversion between Celsius and Fahrenheit. You will get familiar with common Android components and how to interact with them.



(a) Initial App Screen

(b) Convert F to C

(c) Convert C to F

Figure 1, Application Screen Shots

$$C = \frac{5}{9} \times (F - 32) \text{ and } F = \left(\frac{9}{5} \times C\right) + 32$$

Part 1 (50 Points): Using Buttons

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

1. The string values used for the button labels should be read from the strings.xml file and should not be hardcoded in the layout file.
2. Use "Hint" attribute to set the "Enter Temperature" grayed out hint in the Edit Text.
3. Create onClick handlers for each of the buttons:
 - a. Clicking "C to F" should convert the entered value from Celsius to Fahrenheit.
 - b. Clicking "F to C" should convert the entered value from Fahrenheit to Celsius.
 - c. Clicking "Reset" should clear the entered value and reset the UI to match Fig 1(a).
4. Your code should check for special cases such as when no value is entered or invalid number or special characters. In such cases, display a Toast message indicating the error, and prevent the conversion.

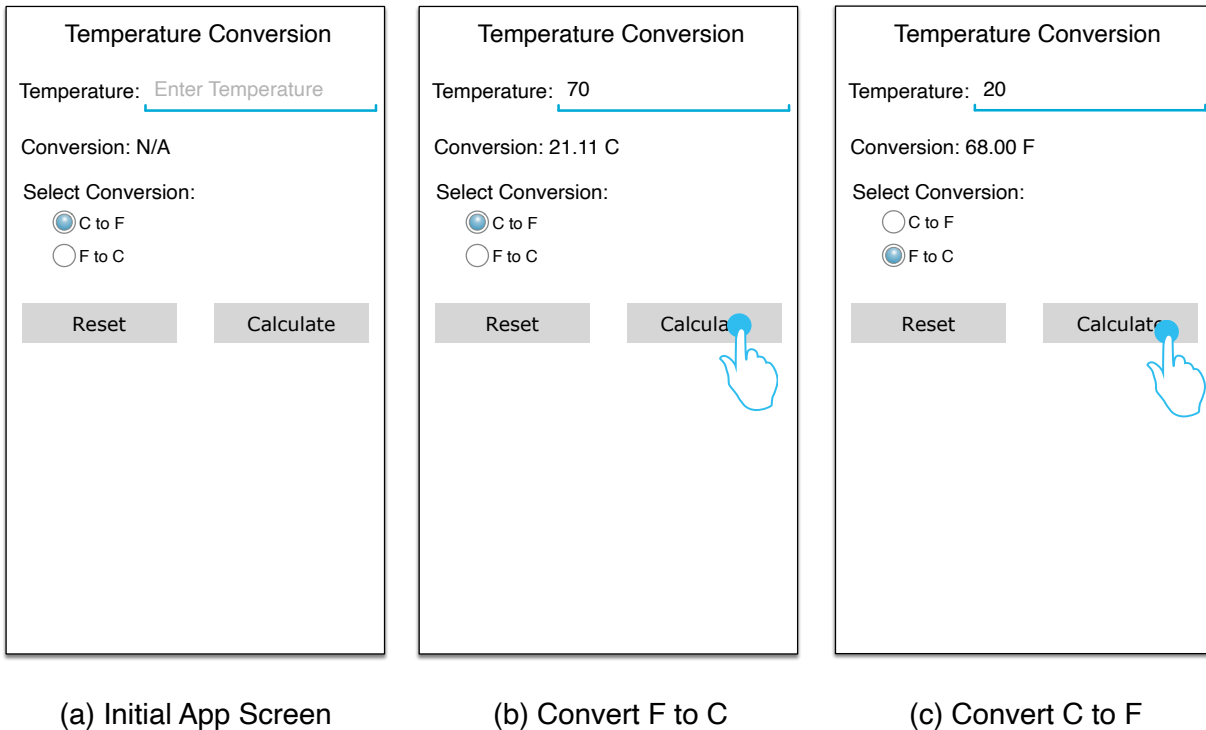


Figure 2, Application Screen Shots

Part 2 (50 Points): Using Radio Buttons

The interface should be created to match the user interface presented in Figure 2(a). You will be using layout files, and strings.xml to create the user interface. The layout XML file can be modified through the raw xml, or through the GUI tools provided within Android Studio. To build the UI, please follow the following tasks:

1. The string values used for the button labels should be read from the strings.xml file and should not be hardcoded in the layout file.
2. Use "Hint" attribute to set the "Enter Temperature" grayed out hint in the Edit Text.
3. Create onClick handlers for each of the buttons:
 - a. Clicking "Calculate" should:
 - i. Retrieve the number entered in the EditText
 - ii. Retrieve the selected RadioButton to decide on which conversion to perform
 - iii. Calculate the converted temperature.
 - iv. Display the value as shown in Fig 2(b) or Fig 2(c).
 - b. Clicking "Reset" should clear the entered value and reset the UI to match Fig 2(a).
4. Your code should check for special cases such as when no value is entered or invalid number or special characters. In such cases, display a Toast message indicating the error, and prevent the conversion.