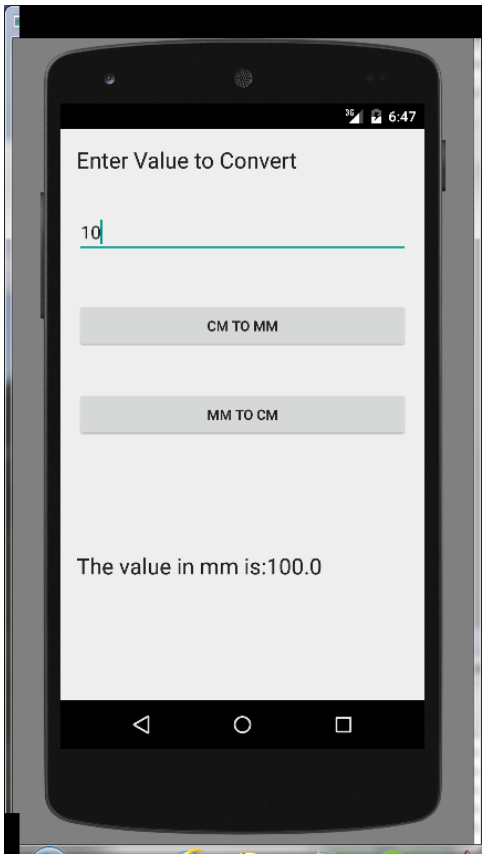
Week 4
Tutorial A

Exercise 1

The applications you test on Virtual Devices can be uninstalled in the same way you can uninstall them on a real device. Try uninstalling the “Reminders” application from the AVD.

Exercise 2

Design and create a simple application which allows users to enter a value and then converts that value from either cm to mm or from mm to cm. The conversion type can depend, for example, on which button a user decides to press. An example UI is illustrated below, along with typical user input and application output:



Hints:

- Remember to give each button a unique id and to set the on click listener for each button.
- You need to consider how to read in numeric data from the user interface. One option is to read that data in as a String and then convert it to double. The Java API has built in methods for achieving this. For example you can use:

```
String name = editText.getText().toString();
double value = Double.parseDouble(name);
```

- You will need to pass the text view a String in order to display the result of your conversion. One way this can be achieved is using the String.valueOf() method, as illustrated below:

```
// hard coded for demonstration
double number = 10.5;
String numberAsString = String.valueOf(number);
```

- How can you ensure your application does not crash is invalid user input is given? For example, your application should not crash if the user enters characters instead of a number. Extend your application code so that it is robust when users give invalid inputs (Look at try-catch statements from this week's lecture).

Exercise 3

Open the Android Studio solution called MyTrackSession2A, available on SunSpace and execute it using an Android Virtual Device.

This is a very basic application with minimal UI components which illustrates how to create instances of an object and store those instances within an ArrayList. Navigate to the java directory in your project and you will notice there are two java files, specifically:

- MainActivity.java.
  This class is similar to the classes that you built during tutorials last week. You will see the same overridden method headers, such as onCreate().
- AudioTrack.java.
  This class represents audio tracks. When an object is created (by calling its constructor), the track name, artist name, album title and price are stored.
  You will notice that the track contains get and set methods for its private member variables, along with a toString() method, which returns the object details in String format. You will not use all of these methods, but they have been coded to give you a more complete picture of how you might create similar classes of your own.

Navigate through the code in both MainActivity and AudioTrack. Make sure you understand the following:

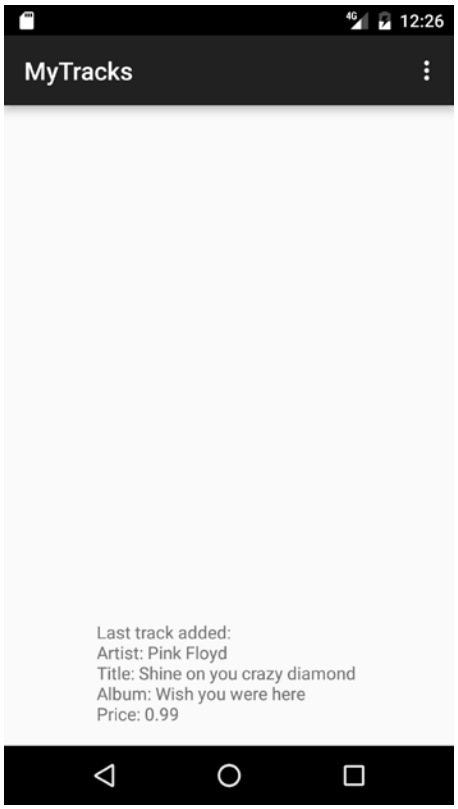
- Where instances of AudioTrack are created.
- Where a TextView object is declared in MainActivity and instantiated using findViewById.
- Where the TextView object is updated at run time using the setText() method.

If you do not understand any of this then ask your instructor for further explanation.

Exercise 4

Adjust the Activity view so that it looks like the UI illustrated below. You may experiment with layout properties if you wish to make a more personalised UI.

Hint – I have added 4 Text widgets (TextView), 4 Text fields (EditText), and one Button to the existing code. Think carefully about keeping your component ids consistent. This will help you in Exercise 5.



Exercise 5

Implement code in MainActivity which adds the item entered by the user to the list of AudioTracks when the "Add Track" button is clicked and then updates the details of the last track added.

This will involve:

1. Declaring each component in your main activity class, eg

Button submitButton;

2. Instantiate each component inside the onCreate() method, eg

submitButton = findViewById(R.id.button_submit);

3. Capturing user input and responding to it, using one of the approaches to event handling described in the lectures. For example, you could implement View.OnClickListener and overriding the onClick() method or you could declare the method for handling button clicks directly in the XML properties.

4. Adding a new track instance to the list of tracks and updating the latest track description

Exercise 6 (optional)

If you have finished these exercises and would like to be more challenged, implement methods which allow a track to be displayed, edited or deleted (by index number). Expose this functionality through your UI.