Worksheet 3

CS 501 – Mobile Application Development

Fall 2022

Android Activities: Lifecycle, Resources, Localization, Bundles, Multiple Activities, Intents, Saving State

Date: 9/26/2022

Team Members: Jinpeng Lyu, Alex Wang, Lesley Chen, Tiffany Chen, Sangjoon (Nick) Lee

Part 1

Activity Lifecycle Revisited, Saving State.

1. **What is the difference between Early and Late Event Binding? Provide 2 examples of when you might prefer to use Early Binding.**

   Early binding is a compile time process while Late Binding is a run-time process. In Early Binding, class information is used while in Late Binding, the object is used to resolve method calling. An example of Early Binding is method overloading, and an example of Late Binding is method overriding.

   Here are some examples of when we might prefer to use Early Binding:

```java
// Java program to illustrate the concept of early and late binding
class Vehicle
{
    void start() {
        System.out.println("Vehicle Started");
    }

    static void stop() {
        System.out.println("Vehicle Stopped");
    }
}

class Car extends Vehicle
{
    @Override
    void start() {
        System.out.println("Car Started");
    }

    static void stop() {
        System.out.println("Car Stopped");
    }
}

class Main
{
    public static void main(String[] args)
    {
        // `Car` extends `Vehicle`
        Vehicle vehicle = new Car();

        vehicle.start();
        vehicle.stop();
    }
}
```

The output of this program is: Car Started and Vehicle Stopped.

In the above java code, it has two classes: Car class extends the Vehicle class. Each of them has two methods—start() and stop(). The start() method is non-static while the stop() method is static. When we call those two methods using parent class reference, the start() method of the Car class is called while the stop() method of the Vehicle class is called. The binding of the stop() method happens at the compile time (Early Binding). **Using Early Binding helps to maximize the integrity preserved for the parent class since methods can't be overridden.**

The second example of Early Binding is:

```
System.IO.FileStream FS ;
FS = new System.IO.FileStream("C:\\temp.txt",
System.IO.FileMode.Open);
```

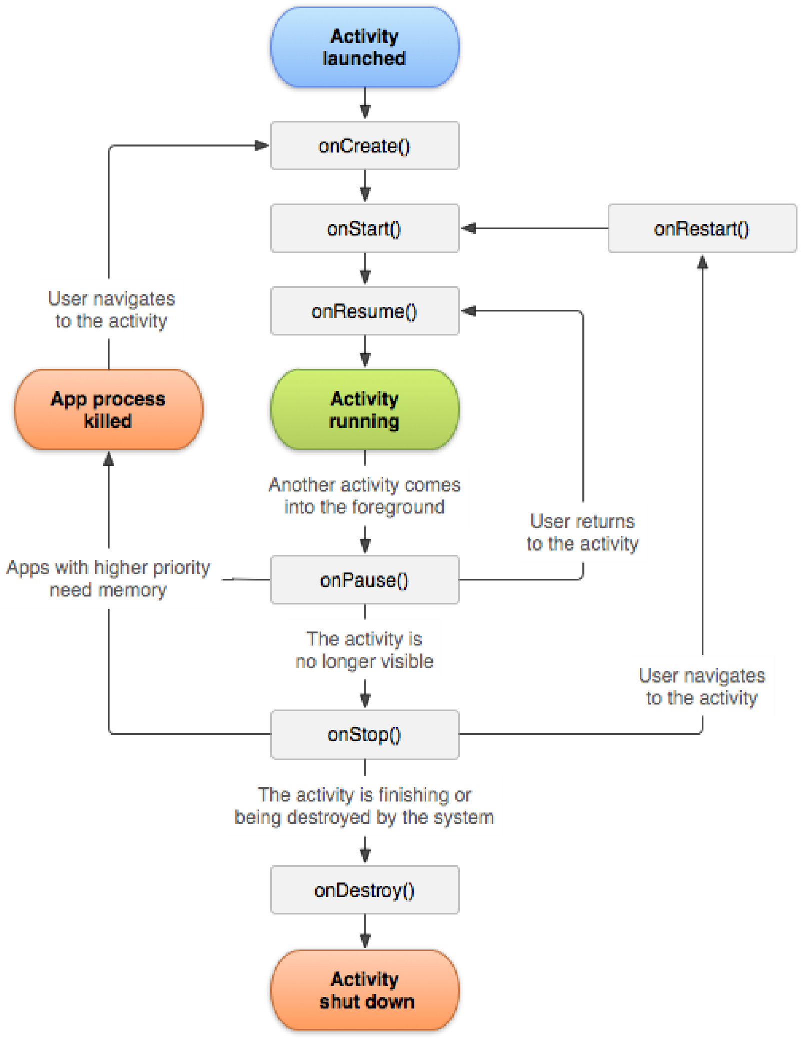
In the above code, we create a variable FS to hold a new object and then assign the file path(object) to FS. Since we are using a declarative method to assign the object to FS, the type of the new object is already known before we execute the program in run-time. **The advantage of doing this is that the compiler can ensure that the function will exist and be callable during run-time. The compiler also guarantees that the function is taking correct parameters.**

2. **[✓ True] or False: A single event handler can be shared with multiple views.**

   If so, how?

   By implementing the onClickListener and overriding the onClick method

For the next few questions, consider the Activity Lifecycle Diagram which shows some new lifecycle events.



Activity launched
onCreate()
onStart()
onRestart()
User navigates to the activity
onResume()
App process killed
Activity running
Another activity comes into the foreground
User returns to the activity
Apps with higher priority need memory
onPause()
The activity is no longer visible
User navigates to the activity
onStop()
The activity is finishing or being destroyed by the system
onDestroy()
Activity shut down

3. **True or False: The Android Framework can kill your application if it's low on memory when it stops.**

4. **True or False: When an Activity is killed, your application is killed.**

5. **Which Activity Lifecycle event(s) should you call upon, when storing and recovering application state, eg, when the phone is rotated.**

   onCreate(), onResume()

6. **Android utilizes the** Bundle **class, sort of like a Hashmap of key/value pairs, to store and retrieve Activity State Information.**

7. **What is a layout in Android?**
   In Android, a layout organizes and arranges components/Views.

8. **True or False: In order to render Views, an Activity must have a layout as its base element (in the XML).**

9. **True or False: A layout can exist within another layout?**

10. **Why is it important to Call super() when overriding base Activity Events?**

    It is important to call super() when overriding base Activity Events because it saves the view hierarchy and allows for reusability.

Part 2

```
Life Cycle Events to Track:
    •   onCreate(..)
    •   onStart(..)
    •   onRestart(..)
    •   onResume(..)
    •   onPause(..)
    •   onStop(..)
    •   onDestroy(..)
```

Implement a program with a blank Activity that logs the Activity Lifecycle as you interact with both the Activity and the device. Log the events listed to the right. ***Use the official Activity Lifecycle from Android to identify the key lifecycle events.*** **Submit the answers here and links to the programs on GitHub.**

Link to Github: https://github.com/sj0726/CS501_Worksheet3_Problem2

2a. What events are called when the App first starts up?
onCreate() ⟶ onStart() ⟶ onResume()

2b. What events are called when the user clicks the home button on the device and then "restarts" the App?
onPause() ⟶ onStop() ⟶ onRestart() ⟶ onStart() ⟶ onResume()

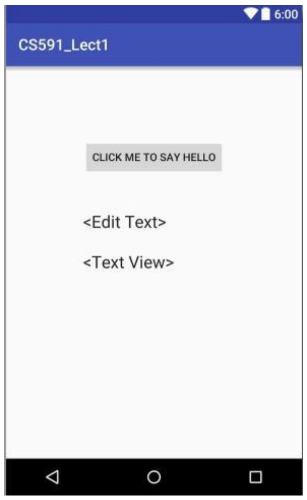
2c. What events are called when the user hits the back button, and then "restarts" the app by clicking on its icon?

onPause() ⟶ onStop() -> onDestroy() -> onCreate() -> onStart() -> onResume()

2d. What events are called if you rotate the phone, switching between portrait and landscape then back to portrait?
onPause() -> onStop() ⟶ onDestroy() -> onCreate() ⟶ onStart() ⟶ onResume() ⟶ Activity running ⟶ onPause() ⟶onStop() ⟶ onDestroy() -> onCreate() ⟶ onStart() ⟶onResume()

2e. What events are called if you open another application and then switch back to the app?
onPause() -> onStop() -> onRestart() ->onStart() -> onResume()

2f. Now Add 3 views to your Activity. When the Button View is clicked "Hello" should appear in both EditText and the TextView. The Activity lifecycle will not change, but you may notice something *odd* about the state of the Text in the EditText/TextViews as you interact with the device.

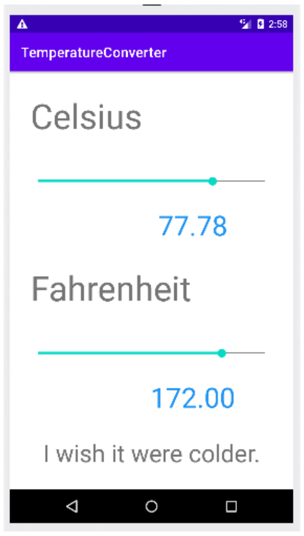


2g. Describe what happens to the "Hello" text inside EditText & TextViews as you leave and reenter your app in the different ways described above.

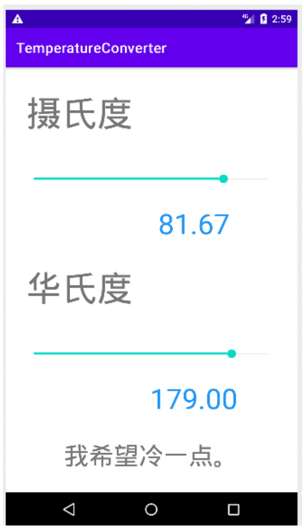
When User hits the back button and then "restarts" the app, the "hello"s in the text fields are reset. The text in textView resets whenever orientation is changed, but editText retains the "hello". If User opens another application and then switches back to the app, both "hello"s are still visible.

Part 3: Temperature Converter, both ways.

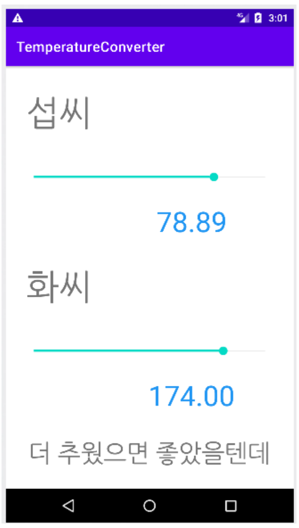
Link to Github: https://github.com/AlAuB/W3_P3
Languages: English, Chinese, Korean



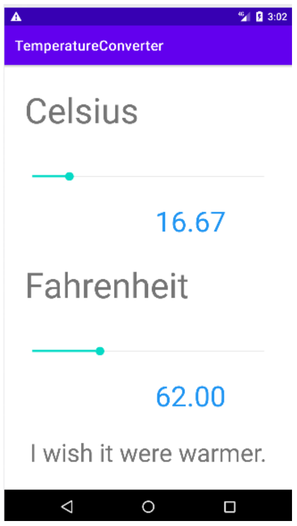
I wish it were colder.



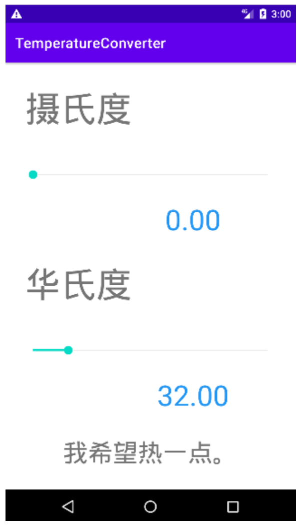
我希望冷一点。



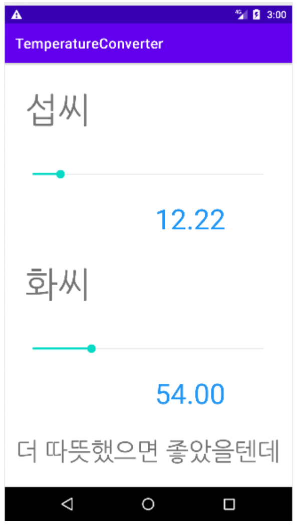
더 추웠으면 좋았을텐데



I wish it were warmer.



我希望热一点。



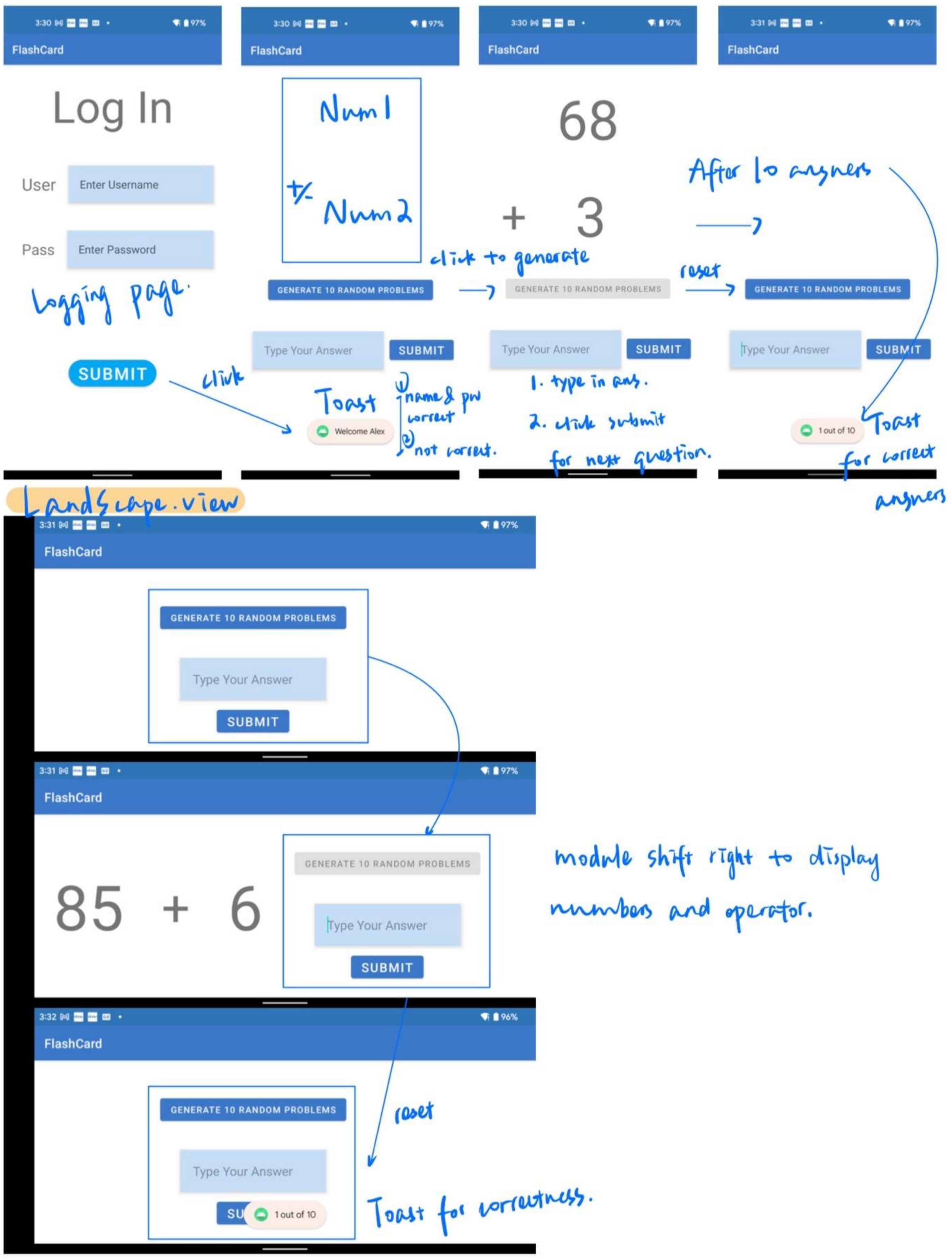
더 따뜻했으면 좋았을텐데

Part 4 - Flash Card App

Github Link: https://github.com/AlAuB/W3_P4

(Username,Password): (Alex, 12), (Lesley, 34), (Louis, 56), (Nick, 78), (Tiffany, 90)

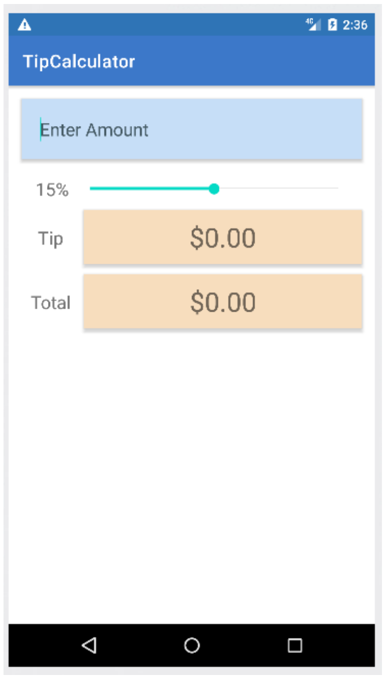
Storyboard:



Part IV: Tip Calculator

Github Link: https://github.com/AlAuB/W3_P5

Screenshots:



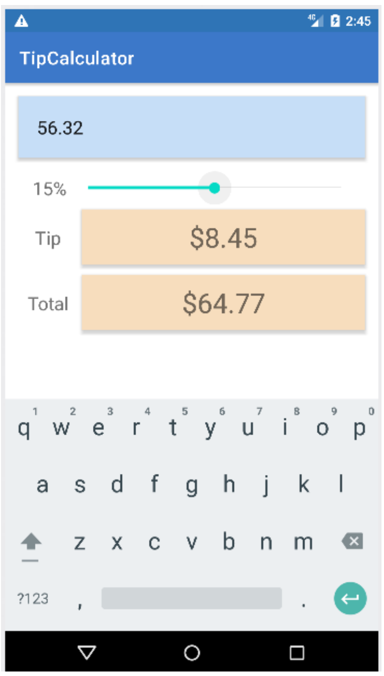
TipCalculator

Enter Amount

15%

Tip $0.00

Total $0.00



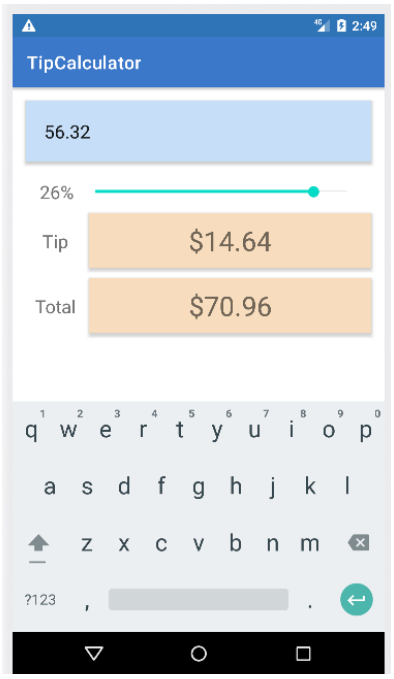
TipCalculator

56.32

15%

Tip $8.45

Total $64.77



TipCalculator

56.32

26%

Tip $14.64

Total $70.96