

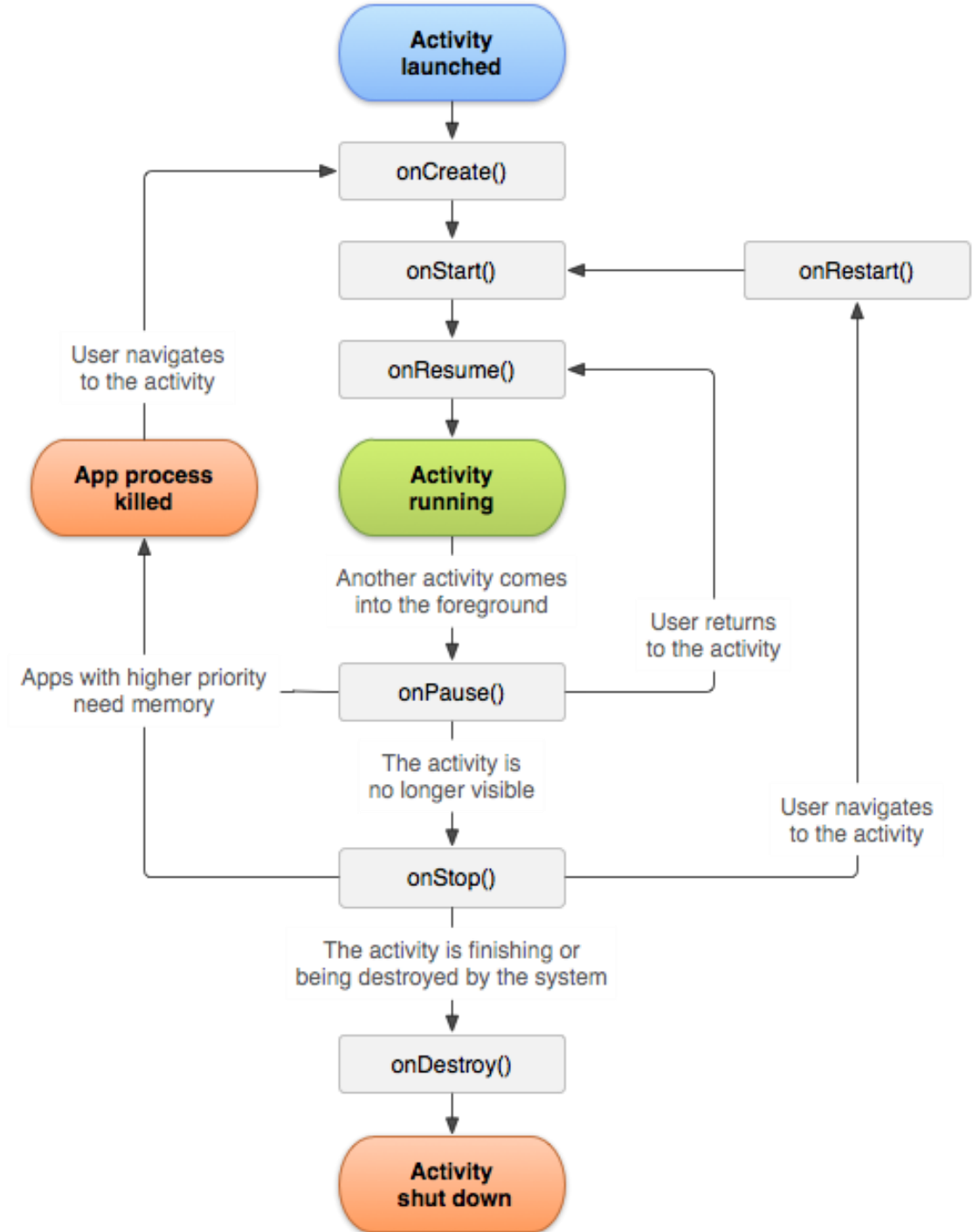


Lab 4 – Activity Lifecycle & Stopwatch

KUAN-TING LAI

2018/10/8

Activity Life Cycle



Layout Types

- Linear layout – Vertical or horizontal
- Relative layout
- Web view

Linear Layout



Relative Layout



Web View

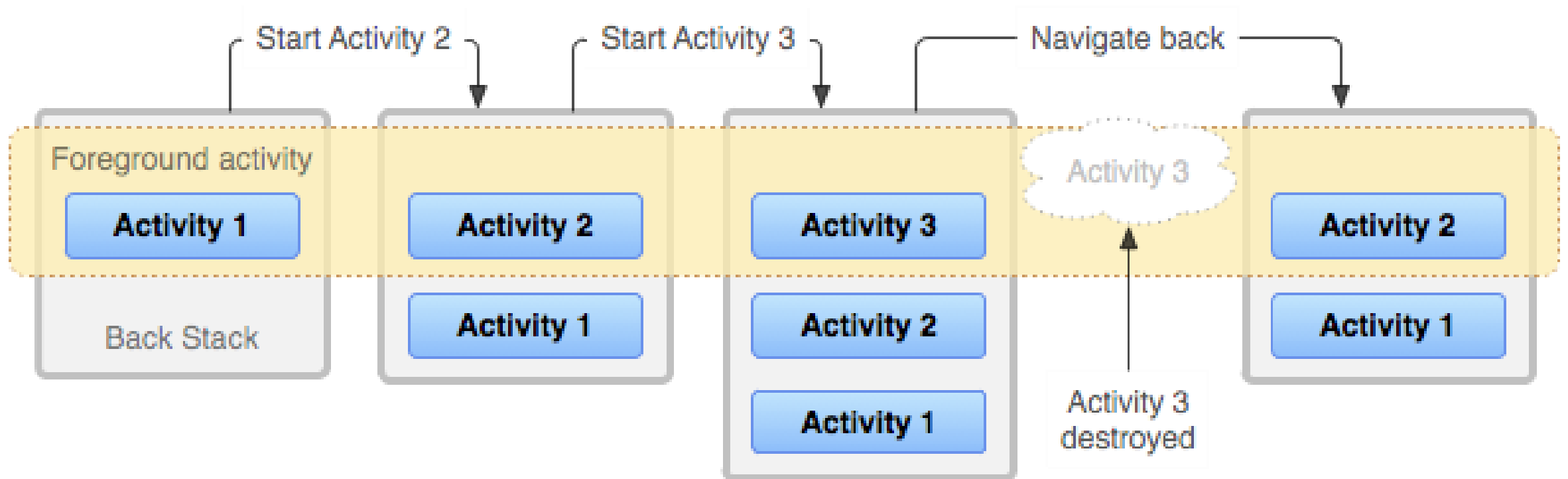


Managing the Activity Lifecycle

- onCreate()
 - Must implement!
- onStart()
 - Called after onCreate()
- onResume()
 - APP regains focus
- onPause()
 - APP loses focus
- onDestroy()

Tasks and Back Stack

- Stack -> last in, first out



0:00:00

START

STOP

RESET

Today's Lab - Stopwatch

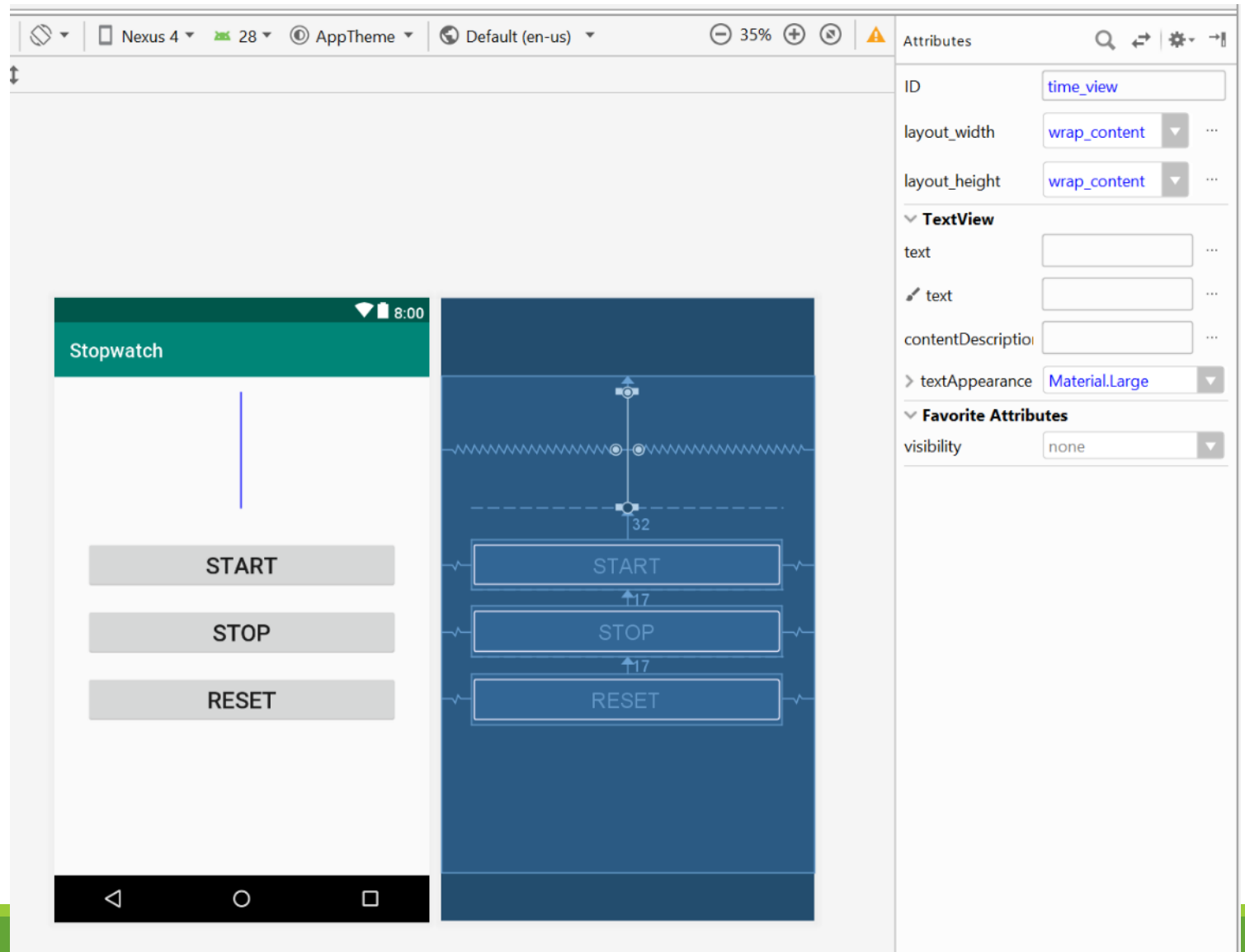
- Activity life-cycle
- Multi-thread

Create a New Project

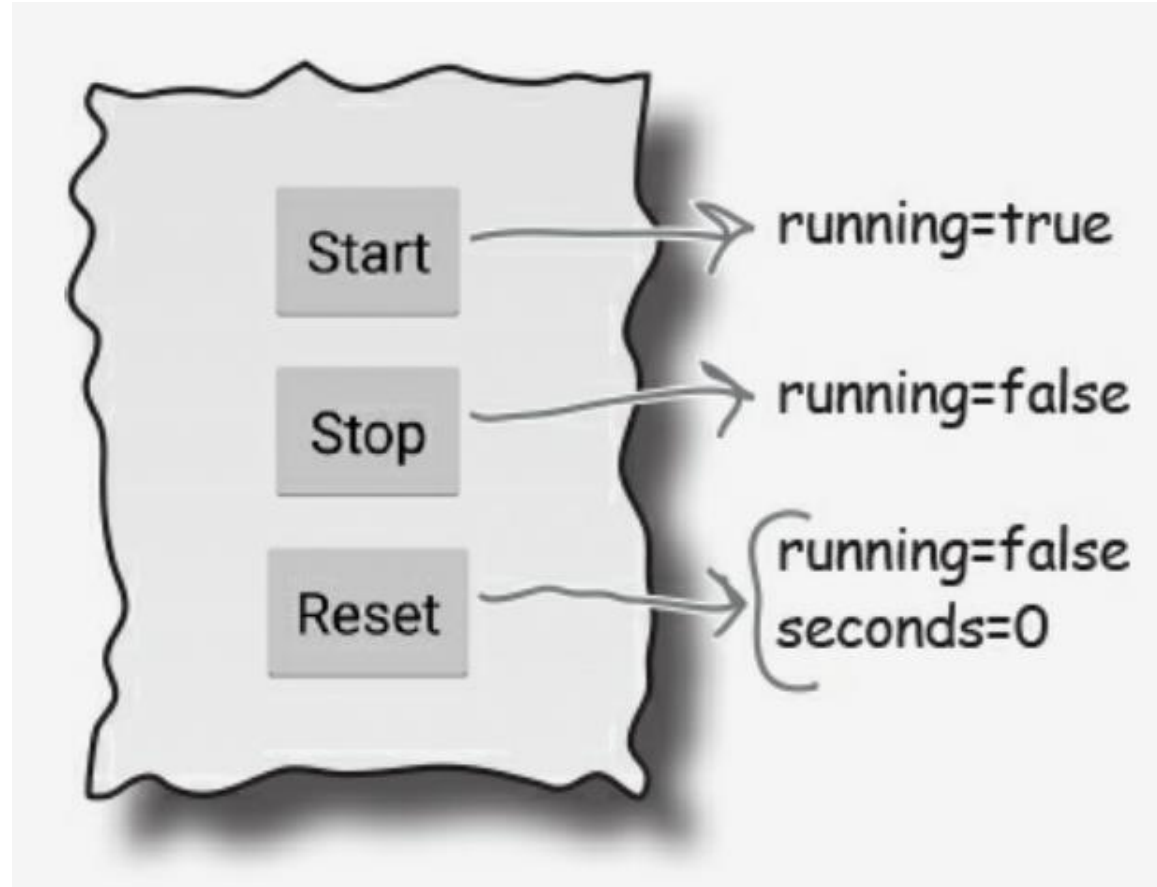
- Create a new project names StopWatch
- Use default API 15
- Select **Empty Activity**
- Use default class name “MainActivity”
- Finish

activity_main.xml

- Add Text View
 - id: time_view
- Add 3 buttons
 - START
 - STOP
 - RESET



Functions of the Buttons



Add functions in MainActivity.java

- runTimer()
- onClickStart
- onClickStop
- onClickReset

```
public class MainActivity extends AppCompatActivity {
    private int seconds = 0;
    //Is the stopwatch running?
    private boolean running;
    private boolean wasRunning;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        runTimer();
    }
    public void onClickStart(View view) {
        running = true;
    }
    public void onClickStop(View view) {
        running = false;
    }
    public void onClickReset(View view) {
        running = false;
        seconds = 0;
    }
    private void runTimer() { ... }
}
```

Add Code in runTimer()

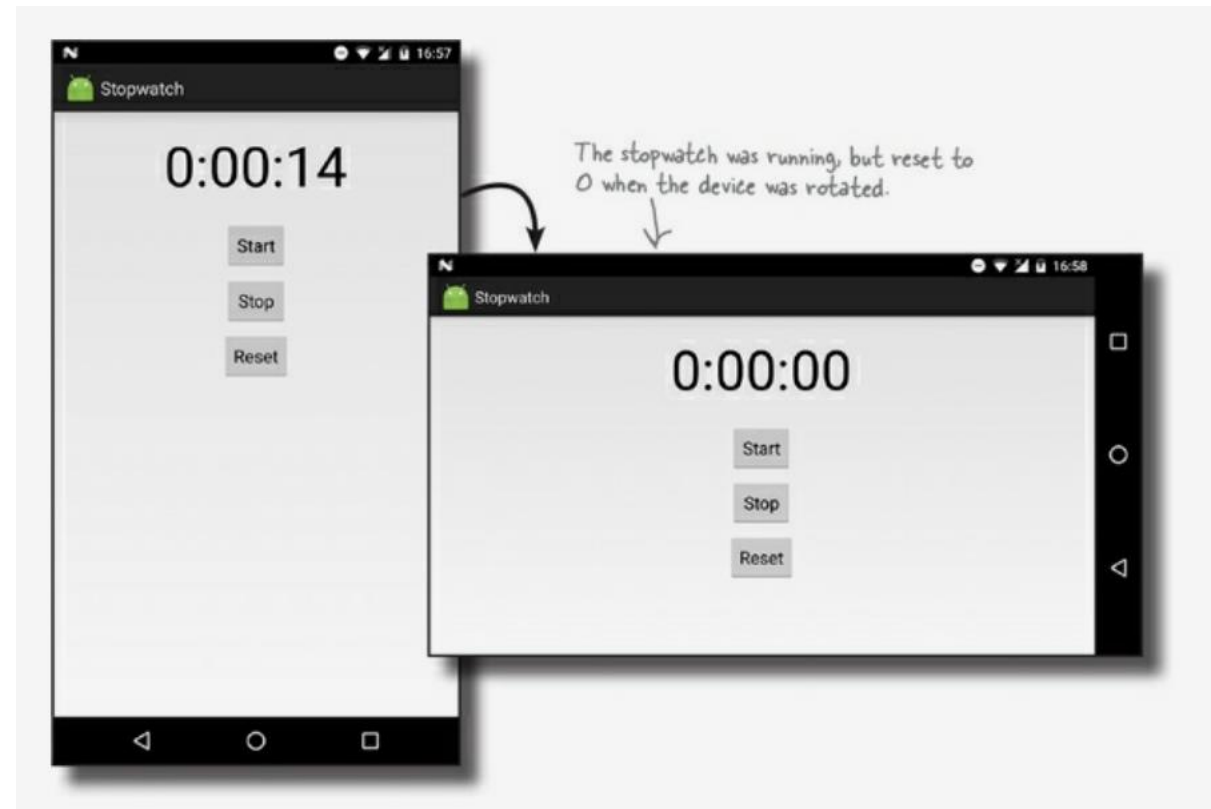
```
private void runTimer() {
    final TextView timeView = (TextView)findViewById(R.id.time_view);
    final Handler handler = new Handler();
    handler.post(new Runnable() {
        @Override
        public void run() {
            int hours = seconds/3600;
            int minutes = (seconds%3600)/60;
            int secs = seconds%60;
            String time = String.format("%d:%02d:%02d", hours, minutes, secs);
            timeView.setText(time);
            if (running) {
                seconds++;
            }
            handler.postDelayed(this, 1000);
        }
    });
}
```

Explanation of Code in runTimer()

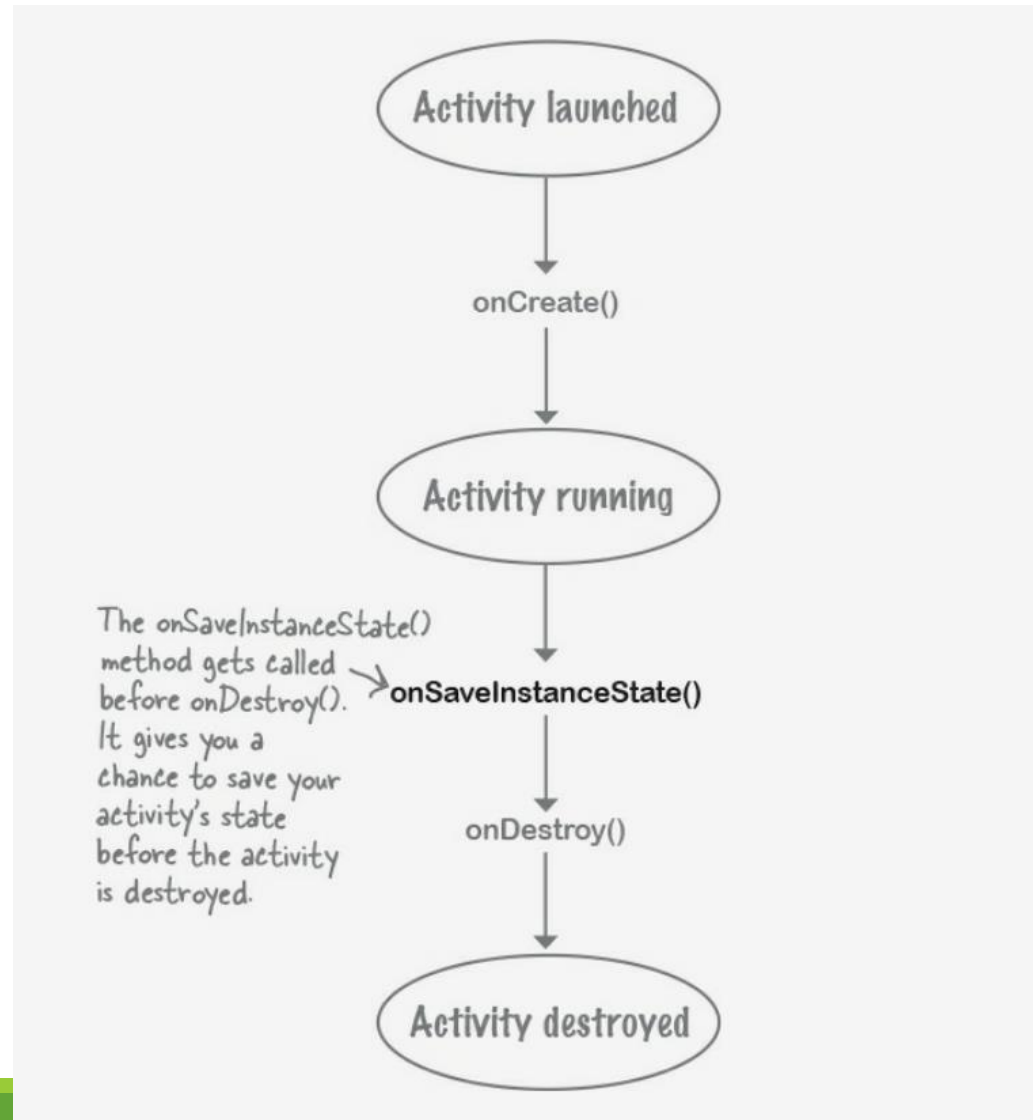
```
private void runTimer() {  
    final TextView timeView = (TextView)findViewById(R.id.time_view);  
    final Handler handler = new Handler(); ← Create a new Handler.  
    handler.post(new Runnable() { ← Call the post() method, passing in a new Runnable. The post()  
        @Override                               method processes code without a delay, so the code in the  
        public void run() {                       Runnable will run almost immediately.  
            int hours = seconds/3600;  
            int minutes = (seconds%3600)/60;  
            int secs = seconds%60;  
            String time = String.format(Locale.getDefault(), ← The Runnable run()  
                "%d:%02d:%02d", hours, minutes, secs);        method contains the code  
                                                                you want to run—in our  
                                                                case, the code to update  
                                                                the text view.  
            timeView.setText(time);  
            if (running) {  
                seconds++;  
            }  
            handler.postDelayed(this, 1000); ← Post the code in the Runnable to be run again  
                                                after a delay of 1,000 milliseconds. As this line  
                                                of code is included in the Runnable run() method,  
                                                it will keep getting called.  
        }  
    }  
};  
}
```

A Little Problem

- The time gets reset to zero when rotated



Save Current Seconds



Override onSaveInstanceState()

@Override

```
public void onSaveInstanceState(Bundle savedInstanceState) {  
    super.onSaveInstanceState(savedInstanceState);  
    savedInstanceState.putInt("seconds", seconds);  
    savedInstanceState.putBoolean("running", running);  
    savedInstanceState.putBoolean("wasRunning", wasRunning);  
}
```

Check savedInstanceState in onCreate()

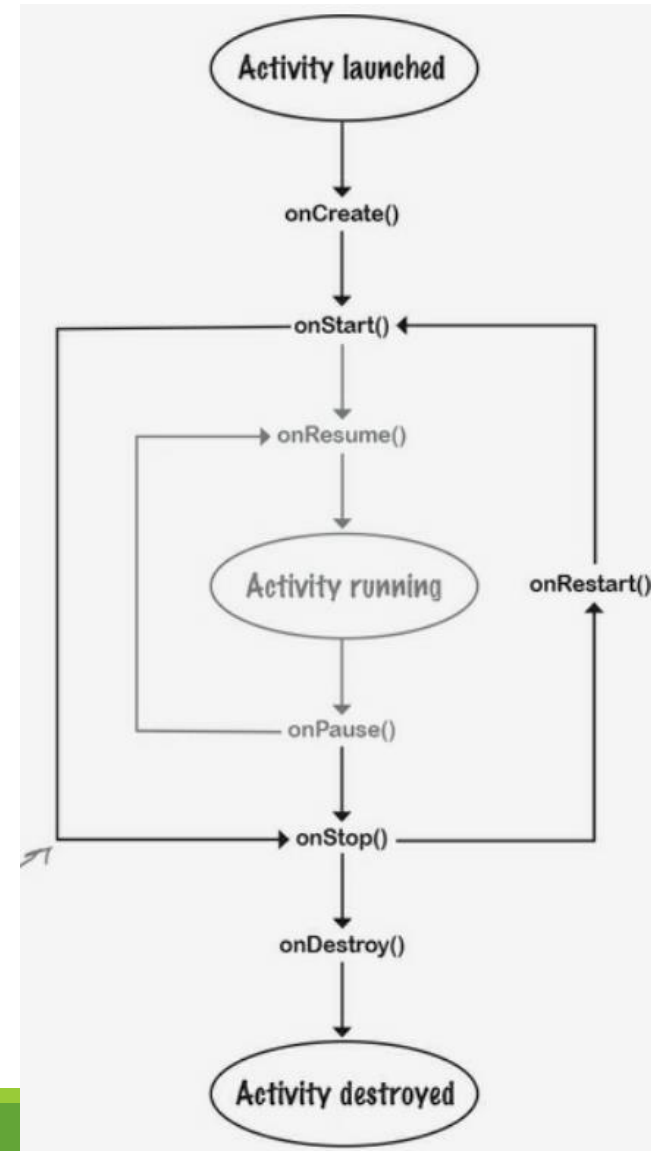
@Override

```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);  
  
    if (savedInstanceState != null) {  
        seconds = savedInstanceState.getInt("seconds");  
        running = savedInstanceState.getBoolean("running");  
        wasRunning = savedInstanceState.getBoolean("wasRunning");  
    }  
  
    runTimer();  
}
```


Pause Your watch when APP Lost Focus

```
@Override  
protected void onPause() {  
    super.onPause();  
    wasRunning = running;  
    running = false;  
}
```

```
@Override  
protected void onResume() {  
    super.onResume();  
    if (wasRunning) {  
        running = true;  
    }  
}
```



加分題

- 顯示1/100秒(小數點後兩位)