

Software Development for Mobile Devices

Submission for Assignment A10.1P

The problem with the given code is that the text view cannot display to the user as expected. Instead of displaying the value from 3-0 each second, the text view display 0 since the `thread.sleep(1000)` is called on the main thread and thus it stops the textview to display the value. Therefore, the best practice to use threads is to do it asynchronously and ensure that they do not affect to the main thread from updating UI.

```
public void click(View view) {  
    try {  
        for (int i = 3; i >= 0; i--) {  
            Thread.sleep(1000);  
            tx.setText(Integer.toString(i));  
        }  
    } catch (InterruptedException ie) {  
        ie.printStackTrace();  
    }  
}
```

The refactor code for the above code

```
private class LongOperation extends AsyncTask<String, Integer, String> {  
    @Override  
    protected String doInBackground(String... params) {  
        for (int i = 3; i >= 0; i--) {  
            try {  
                Thread.sleep(1000);  
                publishProgress(i);  
            } catch (InterruptedException e) {  
                Thread.interrupted();  
            }  
        }  
        return "Executed";  
    }  
    @Override  
    protected void onPostExecute(String result) {  
    }  
    @Override  
    protected void onPreExecute() {}  
    @Override  
    protected void onProgressUpdate(Integer... values) {  
        tx.setText(String.valueOf(values[0]));  
    }  
}
```

```
}  
}
```

Now as the code running in background thread, it does not affect to the main thread. To update the text view, we just simply invoke a main thread to update the UI.