

Name:

Student ID:

Cohort:

ISTD 50.001 Quiz 3

Date: 1-Dec-2017, Friday

3.15 p.m. to 4.45 p.m.

- Write your name and ID at the top of this page.
- Answer all 5 questions. You can secure a maximum of 100 points.
- For questions 1 & 2, starting code is available.
- For questions 1 & 2, submit your code to eDimension
- For question 3 - 5, write your answers on this exam paper.
- All answers will be manually graded. You may be able to earn partial credit for questions.
- **You are not allowed to use any Internet accessing or communicating device during the exam.**
- **If you need a phone to test your app in Questions 1 & 2, your phone must be on FLIGHT MODE and SILENT MODE, and only the apps in this test should be used on your phone.**
- **You are not allowed to consult anyone inside or outside of the classroom other than the course instructors of 50.001.**
- This exam is open book. You may refer to the course slides / notes and your personal notes.
- You can access Java API reference: <http://docs.oracle.com/javase/8/docs/api/> and <https://developer.android.com>
- You can use Android Studio / other IDE to test your programs.
- Good luck!

1. The value of Pi can be estimated by the following algorithm.

```
1. set count = 0
2. decide the number of iterations, n
3. for n iterations
    a. generate a random number x between 0 and 1.
    b. generate a random number y between 0 and 1
    c. if (  $x^2 + y^2 \leq 1$ ) then increase count by 1
4. The estimated value of pi =  $4.0 \times \text{count}/n$ 
```

For a good estimate of pi, the number of iterations should typically be larger than one million. This means that the calculation could take some time on an android phone.

Write an app that allows the user to estimate the value of pi by entering the number of iterations and pressing a button.

The layout of the app before the button is clicked is as follows:

Quiz3_PiQuestion

number of iterations?

RUN THE ALGORITHM

Estimate of Pi here

You do not need to add widgets to the layout file provided to you. However, the attributes in the current layout are not complete, hence you need to complete them. Furthermore, the user input should be restricted to numbers only.

When the button is clicked, the algorithm is launched. The user interface displays the current estimate of pi at every 100000 (one hundred thousand) iterations. A sample message is: "After 200000 iterations the estimate of pi is 2.95". (Note: It is ok if the actual value of the iterations displayed is not an exact multiple of 100000).

If nothing is entered and the button is clicked, display an **Toast** that contains a single message "Please enter a value".

The algorithm must not be executed on the same thread as the user interface.

When the iterations are completed, a toast message "Iterations are complete" should be displayed. The result of the estimate of pi should also be displayed in an appropriate widget.

Put the entire project in a zip file and submit it to eDimension.

(25 marks)

You will just be asked to submit MainActivity.java.

2. You are a music teacher, hoping to help your very young students remember folksongs from different countries. Hence, you hope to build an app for this.

You are given a file containing a JSON database, which is an array of JSON objects. Each JSON object contains the country name and the name of the folksong.

In your project, you need to place this JSON database file in the appropriate directory. In the starter code, complete the method that parses this file.

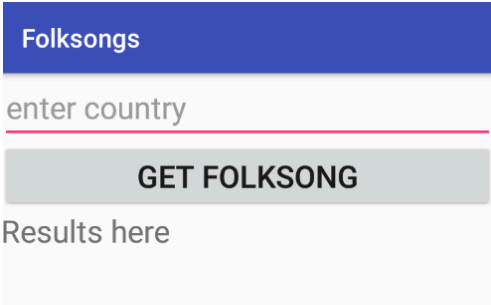
Your app will contain an SQLite database, which is filled with information from the JSON database when created.

Your app then allows the user to key in a country. When the button is clicked, the folksong belonging to that country is displayed.

You may assume that your user will always enter the country name in full, but the case could be in upper or lower case e.g. "Australia", "australia", "AUSTRALIA" or "australiA" could all be entered. There could also be leading or trailing white spaces.

The user merely queries the database, and is NOT able to add content to the database or delete content from the database.

The layout of the app is as follows.



The image shows a mobile app interface with a blue header bar containing the text 'Folksongs'. Below the header is a white text input field with the placeholder text 'enter country'. Underneath the input field is a grey button with the text 'GET FOLKSONG'. At the bottom of the interface is a white text area with the placeholder text 'Results here'.

You do not need to add widgets to the layout file provided to you. However, the attributes in the current layout are not complete, hence you need to complete them.

Put the entire project in a zip file and submit it to eDimension. (25 marks)

You will just be asked to submit the .java files.

4. Study the following code and answer the questions that follow. Only the class definition is shown. The Java file and xml file for this code is given in eDimension.

```
14 public class MainActivity extends AppCompatActivity {
15
16     TimePicker timePicker;
17     TextView textView;
18     final String TAG = "LogcatTag";
19
20     @Override
21     protected void onCreate(Bundle savedInstanceState) {
22         super.onCreate(savedInstanceState);
23         setContentView(R.layout.activity_main);
24         timePicker = (TimePicker) findViewById(R.id.myTimePicker);
25         textView = (TextView) findViewById(R.id.myTextView);
26         Toast.makeText(this, "Welcome", Toast.LENGTH_SHORT).show();
27
28         timePicker.setOnTimeChangedListener(
29
30             new TimePicker.OnTimeChangedListener() {
31                 @Override
32                 public void onTimeChanged(TimePicker timePicker,
33                                         int hour, int minute) {
34                     textView.setText("T=" + hour + ":" + minute);
35                 }
36             }
37
38         );
39
40         Log.i(TAG, "onCreate");
41     }
42
43     @Override
44     protected void onDestroy() {
45         super.onDestroy(); Log.i(TAG, "onDestroy");
46     }
47
48     @Override
49     protected void onPause() {
50         super.onPause(); Log.i(TAG, "onPause");
51     }
52
53     @Override
54     protected void onStart() {
55         super.onStart(); Log.i(TAG, "onStart");
56     }
57
58     @Override
59     protected void onStop() {
60         super.onStop(); Log.i(TAG, "onStop");
61     }
62
63     @Override
64     protected void onResume() {
65         super.onResume(); Log.i(TAG, "onResume");
66     }
67
68     @Override
69     protected void onRestart() {
70         super.onRestart(); Log.i(TAG, "onRestart");
71     }
72
73 }
74
```

a. State the data type of `R.id.myTimePicker` (line 24). (2 marks)

This question is no longer valid. Android Studio 3.0 does not require type casting anymore.

b. At line 24, explain why it is necessary to type cast the object that `findViewById` returns. (4 marks)

c. At line 26, the first input parameter of `makeText` needs to be a `Context` object. Explain why `this` is an acceptable input. (2 marks)

d. At line 28, the `setOnTimeChangeListener` method of the `timePicker` object has one input parameter, which is the code from line 30 to 36. Explain the code from line 30 to 36 as fully as you can. (10 marks)

e. When the app is run, the user then rotates the screen from portrait to landscape orientation. (6 marks)

- i. Briefly describe what you will see in the Logcat and on the screen.
- ii. Hence, state what happens to an activity when the screen is rotated.