Programme Code: TU856
Module Code: CMPU 3026

TECHNOLOGICAL UNIVERSITY DUBLIN

Grangegorman

TU856 BSc Computer Science

Year 3

SEMESTER 1 EXAMINATIONS 2022/23

CMPU 3026
Mobile Software Development

Internal Examiners:
Dr Paul Doyle
Dr Susan McKeever

External Examiner:
Ms Sanita Tifenale

Answer all four questions.
All questions carry equal marks.

Exam Duration: 2 hours.

Q1 **(a)** A developer is choosing between *XML attributes, shared listeners* and *anonymous classes* in order to implement click behaviours for three buttons on a screen.
In your own words, compare these three approaches with regard to (a) Ease of software maintenance and (b) Complexity of implementation.

(12 marks)

(b) Examine the following code snippet in **Figure 1** and explain your answers to these questions:

(i) What does "this" refer to on Line 8 (2 marks)

(ii) How many methods do you think the `OnGesturePerformedListener` interface contains, and why? (3 marks)

(iii) On line 9, what class do you think the getLayoutInflater() method belongs to, and why? (2 marks)

(iv) What is the effect of executing the finish() method on line 14? (2 marks)

(v) Explain the reasons for each @override on lines 5, 19 (2 marks)

(vi) Explain why class name "`Toast`" (as opposed to an object name) appears with the `.makeText` method on line 24. (2 marks)

(Total 13 marks)

```
1    public class GestureTest extends Activity implements OnGesturePerformedListener {
2       private GestureLibrary gestureLib;
3
4       /** Called when the activity is first created. */
5       @Override
6       public void onCreate(Bundle savedInstanceState) {
7          super.onCreate(savedInstanceState);
8          GestureOverlayView gestureOverlayView = new GestureOverlayView(this);
9          View inflate = getLayoutInflater().inflate(R.layout.main, null);
10         gestureOverlayView.addView(inflate);
11         gestureOverlayView.addOnGesturePerformedListener(this);
12         gestureLib = GestureLibraries.fromRawResource(this, R.raw.gestures);
13         if (!gestureLib.load()) {
14            finish();
15         }
16         setContentView(gestureOverlayView);
17      }
18
19      @Override
20      public void onGesturePerformed(GestureOverlayView overlay, Gesture gesture) {
21         ArrayList<Prediction> predictions = gestureLib.recognize(gesture);
22         for (Prediction : predictions) {
23            if (prediction.score > 1.0) {
24               Toast.makeText(this, prediction.name, Toast.LENGTH_SHORT)
25                     .show();
26            }
27         }
        }
     }
```

Figure 1 Code Snippet

Q2 **(a)** Describe in your own words the extent to which the three aspects of the *Model View Controller (MVC) software architecture* are supported in the software structure for a shopping app as described below. Mention any *improvements* you can see to make it more MVC compliant:

The shopping app displays a list of shopping items. The data is stored in a SQLlite local database. The app consists of:

- A class that extends SQLLiteOpenHelper to create the database.
- A class that extends ListActivity, uses the helper class to connect to the database, uses methods from the SQLiteDatabase class to query the database.
- XML layouts for the list and row layout.
- Images files in the Resources directory.
- Event handler methods specified as XML screen layout attributes.

(13 marks)

(b) Looking at the two code snippets in **Figure 2**, answer the following questions, with explanation:

(i) What is the object type of the object called `db`? (2 marks)

(ii) Explain any *two advantages* of the approach of Example 2 over Example 1 for adding data to the database. (5 marks)

(iii) Write **one line of code** that adds a row to the database by calling the method shown in Example 2. Assume that the values will come from three `EditText` objects, ed1, ed2, ed3 and the `edittext` objects have been initialized already using `findViewbyId`. Capture the return value in the code.

(5 marks)

Example 1

```
db.execSQL("INSERT INTO bookmarks (firstName, title, publisher),
‶ + " VALUES('" + firstName] + "', '" + title "', '" + publisher
+ ");");
```

Example 2

```
public long insertPerson(String firstName, String surname,
String city)
{
   ContentValues initialValues = new ContentValues();
   initialValues.put(firstName, firstName);
   initialValues.put(title, title);
   initialValues.put(publisher, publisherName);
   return db.insert(DATABASE_TABLE, null,initialValues);
}
```

Figure 2 Inserting Data

Q3 (a) Android app development is complicated by the variation in hardware and software on end user phones. For *each* of the following variations, discuss how each of them can be addressed when creating an app:

- Variations *in Android API version* on end user devices
- Variety of *screen sizes* on end user devices
- Variety of *screen resolutions* on end user devices

(9 marks)

(b) Explain any four advantages of *native apps* over mobile web apps

(8 marks)

(c) An Android project contains the following `colors.xml` file in the `res/values` project directory. Explain :

(i) Two advantages of defining colours in this way for a project (5 marks)

(ii) Three XMl grammar errors in the file (3 marks)

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
   <color name="opaque_red">#f00000</color>
   <color name="translucent_red>#80ff00<color>
</resources>
</colors>
```

colors.xml file

Q4 (a) An activity implements location tracking. Answer the following:

(i) Explain two parameters available to the developer to control the conditions which determine whether the phone location has changed. (7 marks)

(ii) Explain how the use of *Android lifecycle methods* can be used to control location tracking efficiency (6 marks)

(b) Explain why AsyncTask is currently *deprecated* in Android, and describe how to implement an *alternative* solution for running a parallel activity in the background to prevent overloading the main activity.

(12 marks)