BERGEN UNIVERSITY COLLEGE
Faculty of Engineering

EXAMINATION IN  : TOD 142—Mobile Technology
CLASS           : Computer science, 2nd and 3rd year
DATE            : 19.12.2013

NUMBER OF TASKS : 5
NUMBER OF PAGES : 5
APPENDIXES      : 0

EXAM AIDS       : None
TIME            : 09:00 – 12:00
LANGUAGE        : English

ACADEMIC RESPONSIBLE
TEACHER(S)       : Kent Inge F. Simonsen
                  Martin Lie
EXAMINER(S)      :

NOTES           : Students may answer in English or Norwegian
Problem 1 (Weight 25%)

a) Explain the following important concepts when developing Android applications and give examples where relevant.
   
   - Context
   - Service
   - Resources
   - Manifest File

b) An Activity has the following callback methods.

   ```java
   void onCreate(Bundle)
   void onStart()
   void onRestart()
   void onResume()
   void onPause()
   void onStop()
   void onDestroy()
   ```

   Draw a graph with nodes representing a callback method, and let a directed edge/arrow show in which order the callback methods may be called during the life-cycle of an Activity.

c) Describe how activities can use Intent objects to communicate with other activities within the same applications and with other applications.

Problem 2 (Weight 20%)

a) Explain the concept of a cursor and the role of cursors in SQLite and ContentProviders.

b) Name three of the ways to persistently store data in Android applications and discuss when each of them is appropriate. Give examples.
c) Given a SQLite table, named “pets”, with the following contents (where the first row are the column names):

<table>
<thead>
<tr>
<th>id</th>
<th>name</th>
<th>species</th>
<th>age</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>pete</td>
<td>zebra</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>spot</td>
<td>turtle</td>
<td>128</td>
</tr>
<tr>
<td>3</td>
<td>tweet</td>
<td>cat</td>
<td>1</td>
</tr>
</tbody>
</table>

And the API documentation for the method query(...):

```java
public Cursor query(String table, String[] columns, String selection, String[] selectionArgs, String groupBy, String having, String orderBy)
```

Added in API level 1

Query the given table, returning a Cursor over the result set.

**Parameters**

- **table**: The table name to compile the query against.
- **columns**: A list of which columns to return. Passing null will return all columns, which is discouraged to prevent reading data from storage that isn’t going to be used.
- **selection**: A filter declaring which rows to return, formatted as an SQL WHERE clause (excluding the WHERE itself). Passing null will return all rows for the given table.
- **selectionArgs**: You may include ?s in selection, which will be replaced by the values from selectionArgs, in order that they appear in the selection. The values will be bound as Strings.
- **groupBy**: A filter declaring how to group rows, formatted as an SQL GROUP BY clause (excluding the GROUP BY itself). Passing null will cause the rows to not be grouped.
- **having**: A filter declare which row groups to include in the cursor, if row grouping is being used, formatted as an SQL HAVING clause (excluding the HAVING itself). Passing null will cause all row groups to be included, and is required when row grouping is not being used.
- **orderBy**: How to order the rows, formatted as an SQL ORDER BY clause (excluding the ORDER BY itself). Passing null will use the default sort order, which may be unordered.

**Returns**

- A Cursor object, which is positioned before the first entry. Note that Cursors are not synchronized, see the documentation for more details.

i) What would you expect to be printed to the log by the following code on the next page (you can ignore system generated part of the log messages).
String[] columns = {"id", "name", "age"};
Cursor c = db.query("pets", columns, null, null, null, null, "age");
c.moveToFirst();
while(!c.isAfterLast()){
    Log.w("", c.getString(1), " + " + c.getInt(2));
    c.moveToNext();
}

ii)
Describe what the purpose of the line "c.moveToFirst()" is and why it is needed.

**Problem 3 (Weight 15%)**

a)
Describe the concept of a Layout.

b)
What are the arguments against using absolute positioning and sizes in a Android applications.

**Problem 4 (Weight 20%)**

a)
Mobile applications often have asynchronous behavior. Describe in what circumstances asynchronous behavior is needed, and why.
Service and AsyncTask are two ways to handle concurrency in Android. Describe how each of them works and give examples to illustrate when each of the methods are appropriate.

**Problem 5 (Weight 20%)**

a)

Discuss a few criteria for choosing between native Android applications and HTML- and JavaScript-based hybrid applications.

b)

Mention some of the differences in testing and debugging of hybrid and native applications.

c)

Give a brief overview of the steps involved in the rendering process (from HTML/CSS files to pixels) in web browser engines.

Good luck :)