## **Exercises and Ice Tasks**

## **OPSC7312 (Android) Semester 2**

1. Create a project in Android Studio named prjAsyncFileDemo. Create a CSV file with a hundred words using a comma as a separator. Place the file in your projects data/data folder. Using an inputStreamReader open the file and read the words into a string object or array. Choose a word randomly and display it in a textview. Use the Async class to accomplish this.
2. Repeat the above exercise in another project prjAsyncFileDownload demo. Instead of creating the file, download the file or text from a URL e.g. <http://iheartquotes.com/api/v1/random>?
3. Write an alarm clock service that runs persistently in the background. In this case, your service may need to periodically execute some code to check whether a prescheduled time has been reached so that an alarm can be sounded. To execute a block of code to be executed at a regular time interval, you can use the Timer class within your service.

**Learning Unit 2 – Content Providers – Exercises**

All these examples and further from upcoming chapters is best deployed on your android smartphone for final testing and deployment, so that you can see live data.

**Exercise 4:**

1. Import the project Provider into eclipse. (Copy project into workspace as well)
2. Refactor the project folder name to ContactsBackup.
3. Delete all code that handled earlier versions of Android as well as the managedQuery code. Watch out for the brackets removal.
4. Create a void SaveContacts that will write the Contacts fields to a comma delimited file name ContactsBckUp.txt Write each record on a new line. Use code from the printContacts void. Add a toast message to say the file has been saved. Your imports should occur automatically as you write code.
5. Add a Save button to your GUI. Adjust your GUI XML file.
6. Add an onClickSave event to your java code. Call the SaveContacts method in this code. You may have to move the Cursor c declaration code.
7. Test your program using the Eclipse Emulator as well as your smartphone as an emulator. How would you view the contents of your file that you have created?
8. Adjust your code so that the file is saved on a SD Card.
9. Change the Save button to a save menu item.
10. Add more fields that you may want to backup.

**Exercise 5:**

1. Create a new project named CallLogDemo
2. Adjust your manifest file to add permissions to READ\_CALL\_LOG.
3. Create the GUI XML file using a linear layout with textviews for the CallName, CallNumber, CallDuration and CallDate
4. Write code to read the Caller Name, Called number, duration and call date from the CallLog.Calls provider.
5. Use a CursorLoader and not the managedQuery for the above.
6. Print these fields in the LogCat and bind it to the textviews.
7. You will get an error “Illegal argument … exception Column “\_id” not found. Google this error and explain it.
8. Correct the above error and document it.
9. The Date does not show correctly – format the date to show correctly.
10. Find the Total of all the call durations and display this in a Toast. You should have a menu item to show this.

**Exercise 6:**

1. Import the project ContentProviders into Android Studio (Copy projects into workspace)
2. Change the project name to Elections2014.
3. You need to change the code and the GUI to Add Party results and retrieve party results.
4. Your ElectionProvider class will use a SQLite DB to create a database named Election2015 and one table named PartyResults2015
5. The table will have two fields PartyName (string) and Votes (integer – represented as a percentage)
6. Be careful when you refactor – keep a backup of your original code.
7. Change the Retrieve Party Results button to show “Show Next party”. This button will show only one party and its results at a time/click in a Toast.
8. Add another button “Show winning Party” that will display the party with the highest votes.

**Exercise 7:**

Create a project prjHealthProvider. Create a content provider to store a person’s name, age, blood pressure, cholesterol, etc.

Create another project UseHealthProvider that uses prjHealthProvider to display the health stats.

**Exercise 8: JSON Example**

Create a database and a table in MySQL to store userID, usernames and passwords in a Users table. Add data for 5 users.

Create a php script that will fetch the data from the database. Create an array. Push the fetched data in the array as key=>value format (associative array). Key would be the column name. Print the array in JSON format with the function json\_encode(). Close the database.

Creating Android App for Parsing JSON. Open Android Studio => Create a New Project. Name the project as FetchJSON. Go to your activity\_main.xml. Change the layout to LinearLayout and vertical orientation. Create a ListView for displaying the details that will be fetched. Use JSON parsing to pull the data and show in this listview. Add the internet permission to your manifest.xml.

**Exercise 9: InsertJSON Example**

Create an Android app that will capture a user’s userID, username and password . This data must be sent to a pHp page in a JSON encoded form. The pHp page must insert the data to the same MySQL Database table as in exercise 8. Basically this project is the opposite of Exercise 8.

Exercise 9 : Google Cloud Messaging

Follow the tutorial here:

<http://programmerguru.com/android-tutorial/how-to-send-push-notifications-using-gcm-service/>

(or alternative <http://hmkcode.com/android-google-cloud-messaging-tutorial/> )

<http://sunil-android.blogspot.in/>

<http://examples.javacodegeeks.com/android/core/content/contentprovider/android-content-provider-example/>

<http://androidexample.com/CONTENT%20PROVIDER/index.php?view=examplecloud&scat=33>

<http://www.techotopia.com/index.php/Android_4_App_Development_Essentials>

<http://www.techotopia.com/index.php/Accessing_Cloud_Storage_using_the_Android_Storage_Access_Framework>