# COM1032 Mobile Computing
## Coursework Assignment #2

*"If you hear a voice within you say 'you canot paint,' then by all means paint, and that voice will be silenced."* **–Vincent Van Gough**

### 1. What are the important dates?
Handout Deadline: 27 March, 2019
Submission Deadline: 20 May, 2019
Feedback Deadline: 10 June, 2019
Module Weightage: 60%

### 2. What do we expect in terms of academic conduct and originality?
In a broader sense, academic misconduct involves any attempt by a student to gain an unfair advantage in any kind of assessment. You are encouraged to read the Student Handbook to ensure that you know what this means. Academic misconduct includes all forms of cheating, plagiarism, and collusion. We will take all measures to check your submission against academic misconduct. ***You should be the only contributor to your submission and should not give your code to anyone else, at any time***.

We ask every student to include a text file as part of their submission that has the following statement of originality: "***I confirm that the submitted work is my own work and that I have clearly identified and fully acknowledged all material that is entitled to be attributed to others (whether published or unpublished) using the referencing system set out in the programme handbook. I agree that the University may submit my work to means of checking this, such as the plagiarism detection service Turnitin® UK. I confirm that I understand that assessed work that has been shown to have been plagiarized will be penalized.***"

### 3. What are the objectives of this coursework?
This coursework involves conceptualizing and implementing a mobile application on the Android platform. We give you the freedom to choose the concept and design the kind of app you want to in order to demonstrate your creativity and understanding. The app should meet the following functional requirements:

1. **Demonstrates the use of basic concepts such as Activities, Layouts and Views [30%]**
   This requirement is satisfied by having at least two (or more) Activities which use Views and Layouts to provide a visually consistent experience.
2. **Implements touch and/or sensor (GPS, Accelerometer, Camera etc.) based interaction [25%]**
   A successful Android app requires the access to in-built sensors to provide an engaging experience to its users. Thus it's vital to learn how to incorporate such sensors in your app. You need to demonstrate the use of at least one inbuilt sensor such as touch, position, or orientation etc. Please see the description of Android Sensors app to appreciate the ways we use mobile phone's inbuilt sensors https://play.google.com/store/apps/details?id=com.lpellis.sensorlab&hl=en

3. **a. Demonstrates the use of threads and/or services [25%]**
   This part is designed to test your skills in threading and service management (or process management), which relate to Android OS and are very important to implement efficient apps.

***OR (choice between 3.a and 3.b)***

   **b. Demonstrates persistence using SQLite [25%]**
   Your app should achieve persistence by means of storing and retrieving data using SQLite API.

In addition to above, you are required to write a document (i.e. a ***Report***) that contains:

1. One or more paragraphs describing the purpose of your Android app. **[5%]**
2. One or more paragraphs with suitable screenshots describing how someone can use your app and what different features does it support **[5%]**
3. Screenshots of your app showing how clicking different user interface elements such as a buttons navigates the user among mocked-up screen layouts **[5%]**
4. Description (diagrams) of how your app uses the external services and/or sensors and/or threads and/or SQLite. **[5%]**

   *For example, you can take inspiration from app descriptions provided on Google Play e.g.* https://play.google.com/store/apps/details?id=com.whatsapp&hl=en_GB

## 4. How and what to submit?

Your submission should be a single .zip file named "***urn_android_coursework.zip***", where ***urn*** is your university registration number (uploaded on SurreyLearn). This .zip file should contain the following things:

1. A .zip (named android_app.zip) file of your Android Studio project containing all your code so that it can be opened and run in Android Studio. Your app should run on both the emulator and the tablet.
2. A text file containing declaration of originality statement as above.
3. A pdf file named '***Report***' that has your name, email address along with the name of your app on the top in addition to the four criteria in Section 3.

## 5. How do we evaluate your submission?

We will evaluate your app based on the three functional requirements stated in Section 3 and the report. Specifically we will be looking at the following:

1. Does your app meet the three functional requirements given in Section 2 and to what extent?
2. Does your app work correctly as described in your report?
3. What is the quality of the code in terms of comments, functions and variables naming and layout?
4. How do you utilise object oriented design principles?
5. Is there any clear creative contribution made?

IMPORTANT: It is mandatory to include the references in the report to the code that is not yours. For example if the code comes from online discussion forums. You need to ensure that you understand the code and include appropriate references.

## 6. What can your app look like?

This piece of work gives you ample freedom to explore and demonstrate your creativity. You are free to choose any idea and construct an app that satisfies the three criteria in Section 3. We expect you to have a strong understanding of threads and services, sensor integration and database handling.

The first step would be to think of what kind of app you wish to develop by looking at Android app categories (https://www.statista.com/statistics/200855/favourite-smartphone-app-categories-by-share-of-smartphone-users/). The second step would be to think what functionality your app will implement in order to satisfy the requirements in Section 3. Some app examples are:

1. An app for runners that uses GPS data. https://www.tomsguide.com/us/pictures-story/703-best-running-apps.html. This app can have a UI supported by multiple activities, uses GPS sensor data, possibly use a service running in background updating and checking the current location of the user.
2. A touch based drawing and sketching app. This app can have a service monitoring in-build sensors to provide various features, for example as suggested in the article: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7030191
3. An app that coverts your phone into a remote for PowerPoint presentations (I use this app in my presentations). https://play.google.com/store/apps/details?id=com.pptremotecontrol.android.presenter

## 7. Frequently Asked Questions

1. **I'm not sure if my idea is good?**
   Although, your creativity and extra effort will be appreciated, we expect you to ensure your app meets the three evaluation criteria in Section 3 (grading will be based on the grade descriptor on the next page).
2. **How much time should I spend on my app?**
   Your app should not be a toy Android program like the ones we do in the lab. For example, the simple services and threads related lab exercise that we did in the class. You will have to invest time researching your idea (don't spend too much time on this), studying different concepts (threads, services, sensors, SQLite), deciding the functionality and implementing the app (writing code in Android Studio), and writing the report. We expect you to spend at least 30 hrs on these.
3. **How long should the report be?**
   The max pages you can have is 4 (single-sided).
4. **How will I evaluate your submission?**
   The evaluation will consist of running your app, looking at the code and the report.

## References


1. Activities and lifecycle: https://developer.android.com/guide/components/activities/intro-activities
2. Using sensors in Android app: https://developer.android.com/guide/topics/sensors
3. Threads: https://google-developer-training.github.io/android-developer-fundamentals-course-concepts-v2/unit-3-working-in-the-background/lesson-7-background-tasks/7-1-c-asynctask-and-asynctaskloader/7-1-c-asynctask-and-asynctaskloader.html
4. Services: https://google-developer-training.github.io/android-developer-fundamentals-course-concepts-v2/unit-3-working-in-the-background/lesson-7-background-tasks/7-4-c-services/7-4-c-services.html
5. Writing quality code: https://medium.com/the-andela-way/how-to-start-writing-high-quality-code-at-any-point-of-your-programming-journey-d434cb0ba8ca
6. Concepts of object oriented programming https://catalogue.surrey.ac.uk/2018-9/module/COM1027

## Grade Descriptor

| | 100-81 | 80-61 | 60-41 | 40-21 | 20-0 |
|---|---|---|---|---|---|
| **Activities** | Excellent design and implementation of an App with at least two Activities, Robust use of Intents, Activity Navigation, and Activity State Lifecycle. Goes beyond the material displaying **exceptional integration** of the full range of appropriate principles, theories, evidence and techniques. | Good implementation of an app with at least two Activities, Intents, Activity Navigation, and Activity State and Lifecycle. Goes beyond the material with **very good conceptualisation** which is often original, innovative and/or insightful | Basic Implementation of Two Activities, Intents, Activity Navigation, and Activity State and Lifecycle**. Limited and underdeveloped** critical engagement with the material | Buggy Implementation of Two Activities, Intents, Activity Navigation, and Activity State and Lifecycle. **Fails to demonstrate** sufficient critical engagement with the material | Little or no awareness of appropriate principles, theories, evidence and techniques. |
| **Sensor(s)** | Excellent robust and creative use of inbuilt sensor(s) which goes beyond the material displaying exceptional flair in tackling issues in sensor integration. | Good use of sensor data with very good conceptualisation which is often original, innovative and/or insightful. | Basic implementation of sensor(s), limited and underdeveloped critical engagement with the material. | Incorrect use of sensor (s), fails to demonstrate sufficient critical engagement with the material | No Sensor used |
| **Threads and/or Services** | Excellent demonstration of use of Threads and/or Services to achieve performance goals and also and exceptional application of theoretical and technical knowledge. | Good demonstration of use of Threads and/or Services to achieve performance goals and also and a good use of theoretical and technical knowledge. | Basic application of theoretical and technical knowledge to achieve performance goals using threads and services | Unreasonable use of threads and/or services | No Threads or services used |
| **SQLite** | Excellent demonstration of persistence of app state using SQLite which uses the App lifecycle stages appropriately. Uses multiple tables to store | Good demonstration of SQLite techniques to manage app state and persistence. Supports data multiple tables and queries. | Basic implementation of SQLite techniques supporting a table and basic queries. | Very simple use of database, such as loading data in an array in your code. | No SQLite Support |

| | | | | | |
|---|---|---|---|---|---|
| | data and employs complex queries. | | | | |
| **Code Readability and Quality** | Excellent practices in code writing such as easy to read, well indented, clarity in variable and function naming, supported by useful comments and presence of other relevant attributes. | Good practices in code writing such as easy to read, well indented, clarity in variable and function naming, supported by useful comments etc. | Several aspects from previous column missing in the code. | Really hard to understand and messy code | No effort |
| **Design** | Excellent approach to app design in terms of OO principles in class and object design. Demonstrates a breadth and depth of substantive knowledge that is exceptional and informed by the highest level of scholarship. | Good effort in class and object design, and interaction between objects. Informed by a breadth and depth of substantive knowledge that is comprehensive, accurate, relevant with an awareness of advanced scholarship | Moderate effort in class and object design, and interaction between objects. Demonstrates an adequate breadth and depth of substantive knowledge but with only a few errors or omissions | Below average effort in class and object design, and interaction between objects. Incomplete breadth and depth of substantive knowledge with some errors or omissions | No effort or Demonstrates confusion over the subject Matter. |
| **Layouts and Views** | Excellent implementation supporting landscape mode. Layouts display the Views correctly in any device orientation. | Good response and display in landscape mode however clear minor weaknesses present. | Supporting landscape mode in a minimal form however with major interaction issues. | Doesn't display correctly on landscape mode. | Not supported |
| **Report** | Excellent description of purpose of app, use cases, screen shots and interaction, description of sensors and services. Demonstration of excellent presentation skills. | Good attempt at describing purpose, use cases, screen shots and interaction, and use of sensors and services | Basic attempt at describing purpose, use cases, screen shots and interaction, and use of sensors and services | Ambiguous description of purpose, use cases, screen shots and interaction, and use of sensors and services | No Description |