Encapsulation, Inheritance, polymorphism and abstraction

There are a number of aspects for the complete novice to programming that I will not have the ability to cover. Here are a number of well written tutorials that should cover everything required:

* installing android studio: <https://developer.android.com/studio/install>
* Running your app on a test phone: <https://developer.android.com/training/basics/firstapp/running-app>
* It would also not hurt to pick up some basic java: <https://www.codecademy.com/learn/learn-java> especially learning about the different types of variables

With this provided we assume that you have android studio installed, a device or emulator to test the program and a little exposure to the types of variables that can be encountered in java (although this is not essential).

[insert section written previously]

Classes are how we can divide up code meaningfully into different purposes with different abilities. Imagine, in the car these would be neatly divided up components like the fuel tank or the engine. When building for android it is necessary to inherent a template like design for various components such as templates for running allowing users to interact with a page (a activity) or a template for running functions in the background (a service; though there are many alternatives).

**App structure**

The app will be have four major classes but a number of supporting classes. The major classes which we have to worry about include the: MainScreen, GenerateMessage, EventDetection, DataStorage. It is convention for classes to have their first character capitalised, you are of course free to ignore this convention. The class function inline with their name.

The **MainScreen** will handle the user’s interaction with the first and only screen that they will encounter. This will also determine the flow of the app, it will initiate the participant being informed. Once all the messages to the participant are provided and the participant confirms then then the background logging will start. Finally, the app will also handle requests by the participant to be informed on the amount of time that they have spent on their phone.

**GenerateMessage** will inherit a template/blueprint of the class **MessageTemplate** but change the message generated. Four messages will be displayed demonstrating: the purpose of the app, how the app functions, what security is involved and confirming that the participant consents to take part in the experiment.

**EventDetection** this is a operation which continually runs in the background waiting to detect that an event occurred: that the screen was switched between being on or off. When this is detected we will inform **DataStorage** which then stores the data.

**DataStorage** will store the data securely by utilising the armadillo sharedPreferences (we’ll get to that later). It will also retrieve the data on request.

**Step 1 - start building the app**