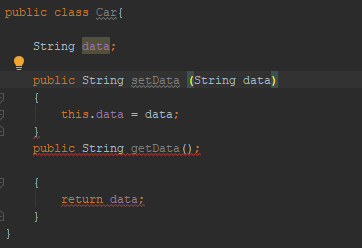
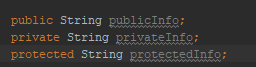
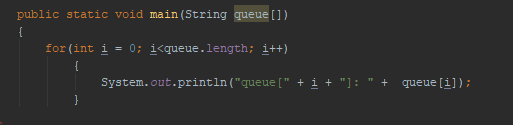
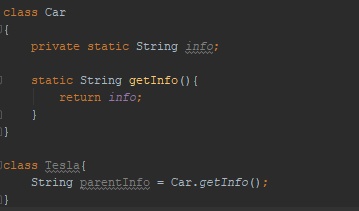
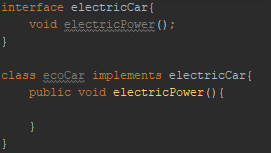
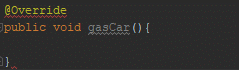
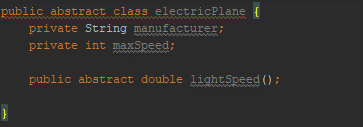
# Advanced Android Labs

## LAB 1

### 1. Study following object oriented concepts

1. Java Object class is the start of class hierarchy in Java. In objects you define the features of the what the class will hold, like states and behaviors, but not all classes are object classes.  
   
2. Classes describe the behavior for one spesific object that it supports (templates/blueprints).  
   
3. Creating of a new instance of a spesific class.  
   
4. These terms relate how we declare variables and methods.  
   Public= means that also everyother class can use it.  
   Private = only that spesific class can use it.  
   Protected = its kinda like private+, that class but also the subclasses can use it.  
   
5. Data members = all the variables contained within classes.  
   Methods = if we would compare to JavaScript, then Methods are what functions are for JS.  
   They are created to classes then called later on to do an action of some sort.  
   
6. This feature almost is completely referred in the name of it. Subclass **inherits** superclasses (parent class) features. Subclass inherits methods and if variables are stated as protected for example.  
   
7. Class that is a collection of methods. It’s an abstract class that has to be inherited by so that it is ”implemented” into a class.  
   
8. This happens when object to take multiple different forms. We use a parent class Car to create class Tesla that extendes to Car but it implements Electric also. Now the class Tesla has multiple features and is considered to be polymorphic.  
   
9. It’s when subclass behavior is defined just spesifically for it but not within the parent class.   
   
10. Abstract classes differ so to others that, they cannot first of all be instantiated. If your class will have an abstract method it has to be declared as such. I feel like this is the hardest to explain and show, but to my understanding Abstract classes are used when you want to pass some details to subclasses and those subclasses will be the ones to actually build a functional thing.  
    

### 3. Study Android fundamental concepts

1. You can use either Kotlin, Java and C++. But technically you can build Android applications with React (react native) and with multiple other sources, but this is completely a different thing.
2. APK means android package. APK is compiled with Android Studio for example so that it can actually be run on a Android device. APK is what EXE is for Windows as an example.
3. Every single Android application runs as a own Linux process. Android Apps are only run when they are needed and are immediately shut if not.
4. Activity = Is what the user will see when displaying the application on his/her device  
   Service = Services run the application in the background for various reason, for example you want to have your spotify running even though you switch to another application.  
   Broadcast receiver = Allow apps to receive notification even though it isn’t running. This will make so that the app is only run when the broadcast receiver gets the right data.  
   Content provider = Its a resource sharing component. For example if an application wants to use your images, then the content provider would be your phone’s build-in gallery.
5. Manifest file contains needed information for multiple sources so that it can be run properly. User permissions, minimum API level for example.
6. These are the files that contain information about how things will be shown (styling), so colors and animations etc.   
   Resource files are needed for how to show everything right on the devices. You will most likely have to have resource files for multiple size of screens and devices, or maybe even for different versions of Android.