Research and answer the following questions:

1. How to implement the 4 observables.

In RxJava an object that implements the Observer interface subscribes to an object of the Observableclass. Then that subscriber reacts to whatever item or sequence of items the Observable object emits. This pattern facilitates concurrent operations because it does not need to block while waiting for the Observable to emit objects, but instead it creates a sentry in the form of a subscriber that stands ready to react appropriately at whatever future time the Observable does so.

1. The difference in the 4 observables.

* Flowable
  + Flowable is typically used when an Observable is emitting huge amounts of data but the Observer is not able to handle this data emission. This is known as Back Pressure.
* Single
  + Single is an Observable that always emit only one value or throws an error. A typical use case of Single observable would be when we make a network call in Android and receive a response.
* Maybe
  + Maybe is an Observable that may or may not emit a value. For example, we would like to know if a particular user exists in our db. The user may or may not exist.
* Completable
  + Completable does not emit any data, but rather is focused on the status of execution — whether successful or failure.

1. How to implement the rxjava map and flatmap.

* For map, you need to have an observable and call the method map. This method is going to try to change the object from which the observable has been made.
* For flatmap you need to do the same, but you need to call the subscriber and pass the object transformer into an observable for the subscriber

1. The difference in map vs flatmap.

Map transforms the items emitted by an Observable by applying a function to each item. FlatMap transforms the items emitted by an Observable into Observables. So, the main difference between Map and FlatMap that FlatMap mapper returns an observable itself, so it is used to map over asynchronous operations.

1. What are hot and cold observables.

Observables are functions that tie an observer to a producer. A cold observable creates the producer and listens to it in the same method. A hot observable closes over the producer.