CMPS 312 Mobile Application Development
Assessment # 1
Deadline 21, Thursday, at 3:00PM

Question 1 [50%]

1. Create a new project and name it **BooksDonation**.
2. Copy the **data** folder under your lab repo **assignments/assignment1** into the root directory of your **BooksDonation** project. The **catalog-books.json** file contains the list of books that you will use to solve the questions listed in the table below.
3. Create a class named **Book** that is going to hold the objects of the **catalog-books.json.**
4. **[Hint:** This is similar to the exercise we did in Lab 3, Part B, reading a JSON file and parsing it]
5. Create an object named **BooksRepo** object in a new Kotlin file named **BooksRepo**
6. Inside the **BooksRepo**
    a. read the **catalog-books**.json and parse them into a list of Book objects.
    b. implement the functions shown in the table below
7. Create a test file called **BooksTest** that tests all **BooksRepo** methods.

getBook(name: String)	Returns the book object if found otherwise "Not found" exception.
getBooksByPageCount(pageCount : Int)	Returns the books with pages >= the pageCount parameters. E.g. Calling the function with pageCount=200 should return all the books with pages >= 200.
getBooksByAuthor(author: String)	Returns all the books authored by that specific author. **Note**: some books have more than one author. You should consider those too and return them as well.
getBooksbyCatagory(category: String)	Returns the books for a particular category. E.g. Calling the function with *category = Programming* should return all the programming books.

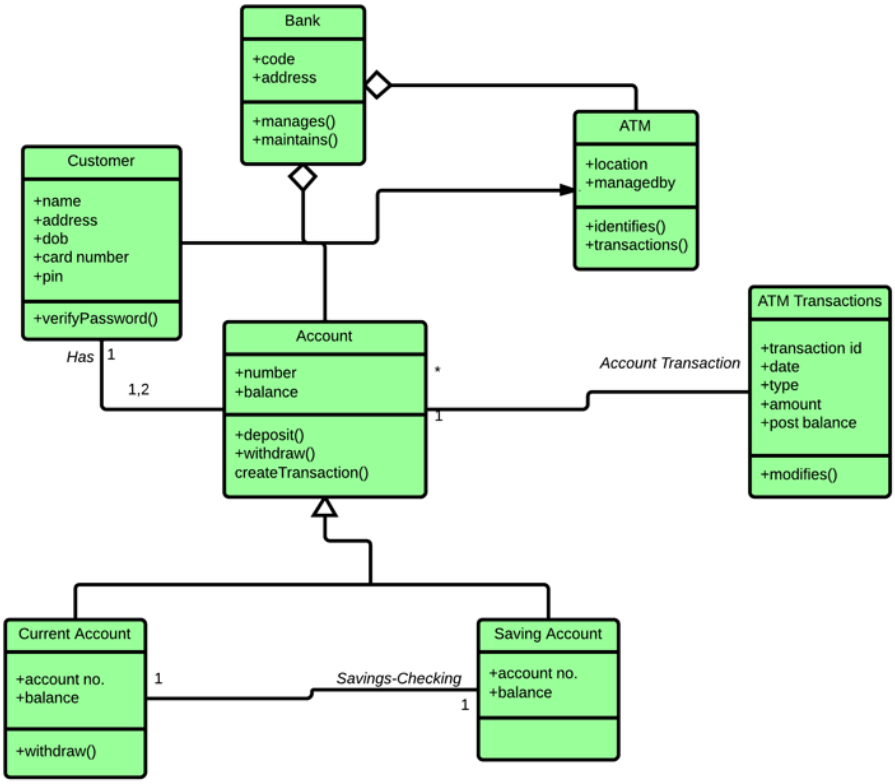
<table>
<tr>
<td>getAuthorsBookCount()</td>
<td>Returns a map that contains the author name and the number of books they have authored. E.g.
<table>
<tr><th>Author Name</th><th>Book Count</th></tr>
<tr><td>James</td><td>2</td></tr>
<tr><td>Ali</td><td>4</td></tr>
</table>
</td>
</tr>
</table>

You should use built-in collection functions (.map, .reduce, .filter, .sort…), in your implementation. You should not use traditional loops such as while loops, for loops etc.… to solve the above questions

Question 2 [50%]

The following relationships are shown in the class diagram belo:

- Customer has one or more Accounts: A customer can have multiple accounts, but each account can only belong to one customer.
- Account has one or more Account Transactions: An account can have multiple transactions, but each transaction can only belong to one account.
- ATM Transaction is associated with one Account Transaction: An ATM transaction is associated with the account transaction that it processed.



Implement the given class diagram for a banking system in Kotlin, and make sure you write the proper validation to prevent invalid operations, such as withdrawing money when the balance is not sufficient or depositing negative amount etc...

Your implementation should include the following:

- A class for each class in the diagram, with the appropriate attributes and methods.
- The relationships between the classes, as shown in the diagram.
- Make sure you write the proper validation . For instance if a person does not have enough balance you should not allow them to withdraw. Also they can not deposit negative n
- Finally write a test class that demonstrates the functionalities of the classes. You should create a list of

Deliverables:

After you complete the lab,

1. Fill out the **Lab1-TestingDoc-Grading-Sheet.docx** file and save it in the Assignments/Assignment1 folder of your repository.
2. Your submission should include the source code for your Android Studio project and the Testing Sheet file.
3. Finally, sync your repository to push your work to GitHub.