**CMPS 312 Mobile Application Development**

**Assessment # 1**

**Deadline Thursday, September 10, 2021 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 be 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. [Hint:** This is similar to the exercise we did in Lab 3, Part B, reading a JSON file and parsing it]
4. Create an object named **BooksRepo** object in a new Kotlin file named **BooksRepo**
5. Inside the **BooksRepo** 
   1. read the **catalog-books**.json and parsethem into a list of Book objects.
   2. implement the functions shown in the table below
6. 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. |
| getAuthorsBookCount() | Returns a map that contains the author name and the number of books they have authored. E.g.   |  |  | | --- | --- | | **Author Name** | **Book Count** | | James | 2 | | Ali | 4 | |

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%]**

Implement the following application that allows users to calculate their BMI and displays one of the following four message [**Underweight, Normal Weight, Overweight or Obesity** ]. You can use the following formulas to calculate the BMI of the user.

|  |  |
| --- | --- |
| * **Underweight** = BMI<18.5 , | * **Normal weight** = BMI between 18.5–24.9 |
| * **Overweight** = BMI between 25–29.9 , | * **Obesity** = BMI of 30 or greater |

Below are the screenshots of the application.

Graphical user interface, application

Description automatically generatedGraphical user interface, application

Description automatically generatedGraphical user interface, application

Description automatically generatedGraphical user interface, application

Description automatically generated

**Deliverables:**

After you complete the lab, fill in the **Lab1-TestingDoc-Grading-Sheet.docx** and save it inside the Assignments/Assignment1 folder in your repository. Your submission should be containing the **source code Android Studio project** and **the Testing Sheet**. Finally, sync your repository to push your work to Github.