CMPS 312 Mobile Application Development
Lab 3-Kotlin OOP and Lambdas

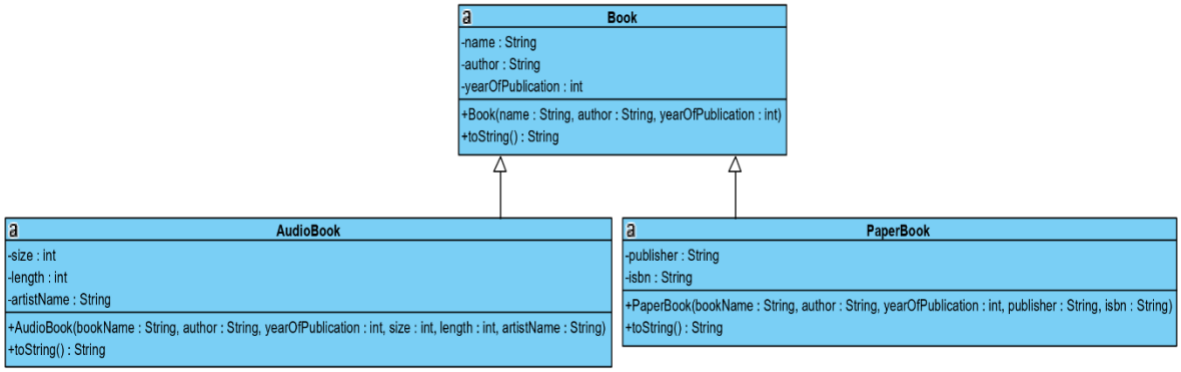
OBJECTIVE

1. Practice Object Oriented Programming (OOP) using Kotlin
2. Read and parse JSON data
3. Practice processing collections using lambdas

PART A – OOP

EXERCISE 1

1. Create an application named **`Books`** with no Activity.
2. Create a package called **`model`**.
3. Implement the following class hierarchy inside the model package.



- The toString() of Book should return **Name, Author, Year of Publication**.
- The toString() of PaperBook should return **Name**, **Author, Year of Publication, Publisher, ISBN**.
- The toString() of AudioBook should return **Name**, **Author, Year of Publication, Size, Length, ArtistName**

The data returned by the toString should be labeled (e.g., ***Name: Ali Baba and the Forty Thieves, Author: Hanna Diyab***).

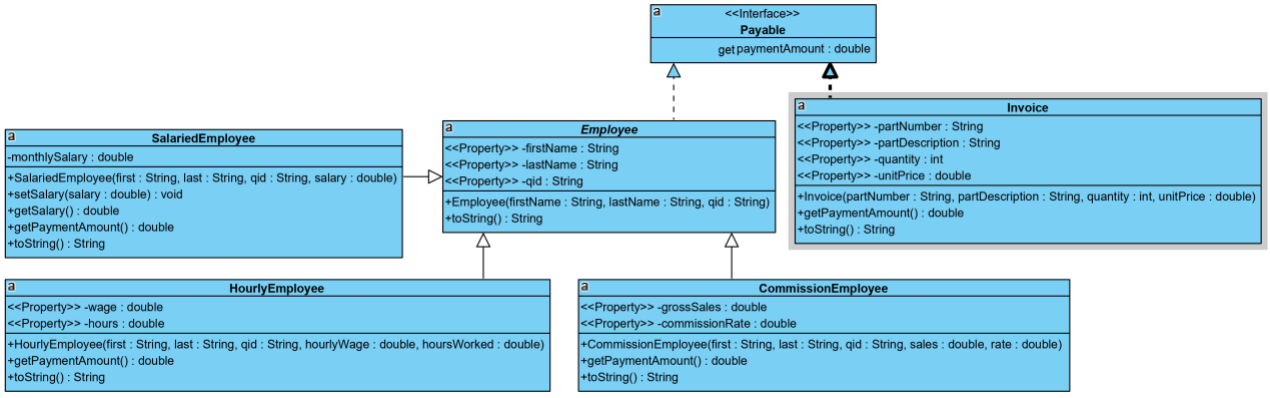
4. Create a main function to test your implementation.
5. In the main function create a List having 2 audio books and 2 paper books.
6. Display the details of each book using the list's forEach method.

Sample Output

```
Book Name : C++
Author Name : John
Year Of Publication : 1/2/2019
Publisher : Oriely
Isbn : 100-11-11


Book Name : Java
Author Name : Mark
Year Of Publication : 1/2/2019
Publisher : NewTimes
Isbn : 100-11-12


Book Name : Android
Author Name : Baaji
Year Of Publication : 1/2/2019
Publisher : Sanford
Isbn : 100-11-13


Book Name : How to get Rich
Author Name : Ali
Year Of Publication : 1/2/2019
Size : 100
Length : 25
Artist Name : Black Panter
```

EXERCISE 2

1. Create an application named "**QU Payroll**"
2. Create a package named **model**
3. Implement the following class hierarchy inside the **model** package



- Note that the amount to pay for HourlyEmployee is *wage * hours*. For CommissionEmployee, it is *grossSales * commissionRate*. For Invoice, it is *quantity * unitPrice*.
- Make sure the **salary**, **rate** and **sales** are all non-negative numbers otherwise display a warning message. [hint: for data validation using init or set methods]

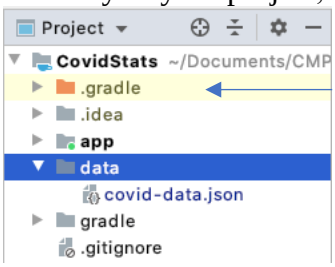
Test your implementation using the main method

```
fun main() {

    // create payable array List
    val payables = arrayListOf<Payable>()

    // populate array with objects that implement Payable
    payables.add(Invoice("01234", "Textbook", 2, 375.00))
    payables.add(Invoice("56789", "USB Disk", 3, 179.95))
    payables.add(SalariedEmployee("Ahmed", "Ali", "111-11-1111", 15000.00))
    payables.add(HourlyEmployee("Fatima", "Saleh", "222-22-2222", 160.75, 40.0))
    payables.add(CommissionEmployee("Samir", "Sami", "333-33-3333", 100000.0, .06))

    println("Invoices and Employees processed polymorphically:\n");

    // generically process each element in array payableObjects using foreach
    payables.forEach { payable ->
        // output currentPayable and its appropriate payment amount
        println("$payable\n")

        //If SalariedEmployee then increase the salary by 10%
        if (payable is SalariedEmployee) {
            val oldBaseSalary = payable.monthlySalary;
            payable.monthlySalary = oldBaseSalary * 1.1;
            println("New salary with 10%% increase is: QR ${payable.getPaymentAmount()}\n");

        }
    }
}
```

```
Invoices and Employees processed polymorphically:

Part Number      : 01234
Part Description : Textbook
Payment Amount   : 750.0

Part Number      : 56789
Part Description : USB Disk
Payment Amount   : 539.8499999999999


First Name :Ahmed
Last Name  :Ali
QID        :111-11-1111
            Payment Amount : 15000.0

New salary with 10%% increase is: QR 16500.0


First Name :Fatima
Last Name  :Saleh
QID        :222-22-2222
            Payment Amount : 6430.0


First Name :Samir
Last Name  :Sami
QID        :333-33-3333
            Payment Amount : 6000.0
```

PART B - LAMBDAS

1. Create an application with and name it "**CovidTracker**"
2. Copy the **covid_data.json** from your lab repo under **Lab 3 folde**r and paste it in the root directory of your project, under a folder called "data" (create this folder yourself)



3. Once you copy the file and open it in Android Studio you might get a warning message showing
   "File size exceeds configured limit (2.5MB). Code insight features not available…"
   To fix the above problem do the following
   a. Go to Help > Edit Custom Properties
   b. Add: *idea.max.intellisense.filesize=999999*
   c. Restart the IDE.

4. Open **build.gradle** and add the following dependencies and plugin and then click on the "**Sync Now** "at the corner of the screen.

```
apply plugin: 'kotlinx-serialization'


//Added for Kotlin Serialization
implementation "org.jetbrains.kotlin:kotlin-stdlib:$kotlin_version"
implementation "org.jetbrains.kotlinx:kotlinx-serialization-core:1.0.0-RC"
```

5. Create a data class called "**CovidStat**" that can hold the following JSON data. You can drive the properties from the below JSON object.

```
{
  "id": 1,
  "country": "United States",
  "continent": "Americas",
  "region": "Northern America",
  "totalCases": 6215592,
  "newCases": 38571,
  "totalDeaths": 187736,
  "newDeaths": 512,
  "totalRecovered": 3456263,
  "newRecovered": 30540,
  "activeCases": 2571593,
  "criticalCases": 15864,
  "casesPer1M": 18759,
  "deathsPer1M": 567,
  "totalTests": 82624841,
  "testsPer1M": 249373,
  "population": 331330464
},
```


6. Create a new Kotlin file named **CovidStats** and implement and test the following functions that return:

- The **total covid death around the world.**
- The total **active cases** for a specific **continent**. Ask the user to give you the continent name and then aggregate and display the total active cases for that continent.
- The **top three countries** with the **highest number of COVID cases in the world.**
- The **top three countries** with the **lowest number of COVID cases in the world.**
- The total critical cases of the neighboring countries sorted by population. Ask the user to give you a country name. For example , if the user puts “Qatar” you should display all GCC neighbor’s and their respective critical cases. Then sort those countries in terms of their population size. **Hint** use region to find the neighbor’s.
- the top three regions in South America with the highest recovery
- the country with the **lowest death** in **specific continent**.