>>اكتب اسمك ورقمك الجامعي في بداية كل class على شكل comment <<

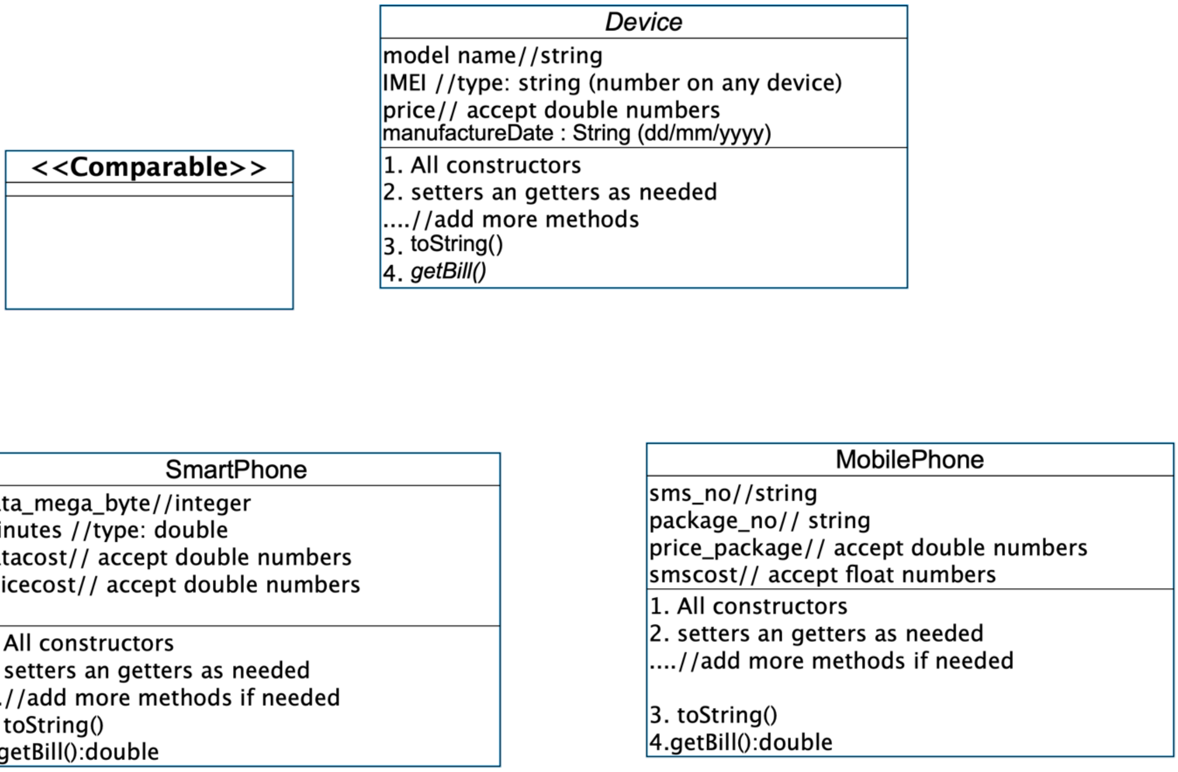
>> تذكير: قم بعمل تسجيل لسطح المكتب <<

>> تذكير: قبل تسليم الفيديوهات، و بعد انتهاء الامتحان، قم بإعادة تسمية الملف بإسمك و رقمك الجامعي و رقم الملف<<

>>يجب تسليم برنامج خالي من الأخطاء وإلا سيتم تصحيح الإجابة من ٥٠ ٪<<

مثال: <<Ahmad Hamed_1220222_Q1>>

Question#1 [40%] **Implement the provided classes using Java, following the discussions in the lectures and labs. Ensure that your implementation complies with Java code conventions.**



<<Comparable>>

Device

model name//string
IMEI //type: string (number on any device)
price// accept double numbers
manufactureDate : String (dd/mm/yyyy)

1. All constructors
2. setters an getters as needed
....//add more methods
3. toString()
4. *getBill()*

SmartPhone

data_mega_byte//integer
minutes //type: double
datacost// accept double numbers
voicecost// accept double numbers

1. All constructors
2. setters an getters as needed
....//add more methods if needed
3. toString()
4.getBill():double

MobilePhone

sms_no//string
package_no// string
price_package// accept double numbers
smscost// accept float numbers

1. All constructors
2. setters an getters as needed
....//add more methods if needed

3. toString()
4.getBill():double

Part A

- Specify the relationship between classes.
- Implement the different constructors for each class.
- Implement the different data members (data fields, setters, and getters methods)
- Implement the **toString** method: return the full object information.

Part B

In the Device class, you must achieve the following:

1. **Ensure that the IMEI meets the following requirements:**
   a. The IMEI consists of **9 digits**.
   b. The **first three** characters of IMEI are letters.

c. The IMEI includes **a mixture of letters and digits**, with all letters **capitalized**.

When reading the IMEI from the user, you must check whether it meets the above requirements before accepting/setting the value. Otherwise, you have to **throw an exception** (**new MyException("The entered IMEI is not valid")**), where **MyException is a custom exception** you have to develop yourself.

Examples of invalid IMEI: **ABX900, ABX900CBz, ABXFGJJKA, 900ABCABC**. These IMEIs are not accepted because they may have a length of fewer than 9 digits, they don't start with three characters, or they contain lowercase letters.

Examples of a valid IMEI: ABX900CBZ

In the SmartPhone class, add the following methods:

getBill () → This method is responsible for returning the value of the bill using the following formula: **data_mega_byte*dataCost+minutes*voiceCost;**

In the mobilePhone class, add the following methods:

getBill () → This method is responsible for returning the value of the bill using the following formula: **sms_no*smsCost+price_package;**

Part C

1. **Write a driver class** called **Driver_YourStudentID**, for example, **Driver_1221089**. This driver **should** test all methods. The driver **must read data from a text file** with the extension '.txt'. The **data will be stored as device objects (smartphone or mobile phone) in an ArrayList**.

   You are free to define the file's structure and give a name to the file as you see fit (You must submit your text file).

2. Implement and call the following methods from the driver class:
   a. A method called **printInfo** accepts the device's **ArrayList** and **prints all the entire object data into the console**.
   b. A method **to sort the device's objects in ascending order** based on their **bill values and print them to the console**. Feel free to provide a name for this method.
   (**Hint**: Use the **Collections.sort(list)** method, which accepts an ArrayList as a parameter and sorts it)
   c. A method that accepts an ArrayList of devices and prints **information for all devices** with **identical prices and manufacture dates**, ensuring that duplicate devices are not printed. (**date format: dd/mm/yyyy**).
   d. A **method to print only the total minutes for smartphone devices**. This method should accept an ArrayList of devices and **print only the total minutes for smartphone devices.**