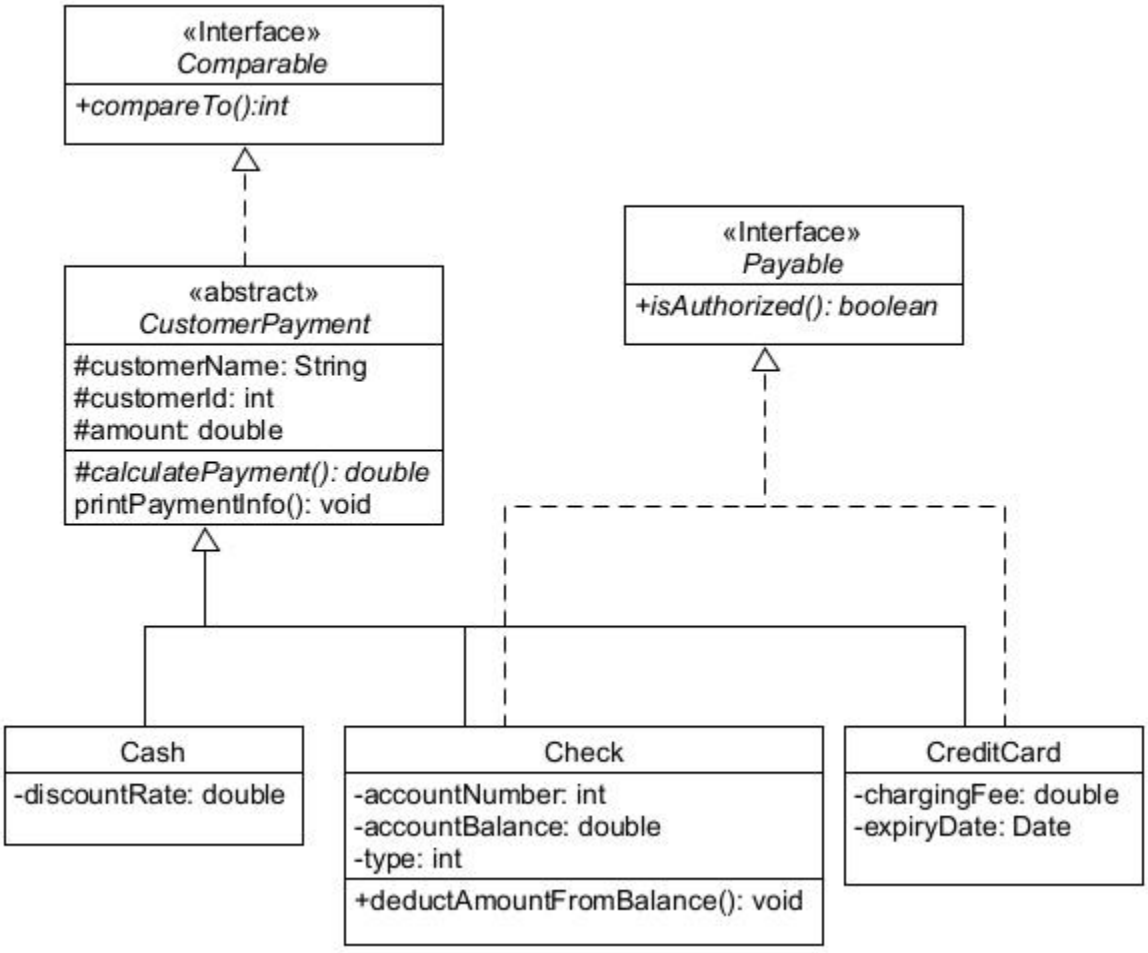
Faculty of Engineering and Information Technology

Computer Science Department

Comp 2310 Project

Individual work project.	**Due Date: Sat. 27/1/2024 by 10:00 pm on *ITC***

Given the following UML diagram:



Please do the following:

1- Create class/interface that implements the classes + interface displayed in the UML displayed above ***exactly as specified*** (same variable names, types, …). Make sure to include no-arg + all field constructors, setters + getters, as well as **toString()** methods in each of the classes created. Please note the following:
   a- method ***calculatePayment***()is implemented differently in each of the concrete classes as follows:
      - in class **Cash** it is calculated as the payment amount entered minus the discountRate percentage (e.g. if amount is 200.0 and ***discountRate*** is 11.0 then payment value is 178.0)
      - in class **Check**, payment is the same as the amount entered.
      - in class **CreditCard**, payment is the amount plus the ***chargingFee***.
   b- method ***printPaymentInfo()*** prints the properties and the calculated payment by calling both the ***toString()*** as well as the ***calculatePayment()*** methods.
   c- class **CustomerPayment** is Comparable based on the value returned by the method ***calculatePayment()***.
   d- **Check** class has a variable called type (int) which is set using one of three constant values that should be defined in the class (CASHIER=1, CERTIFIED=2, PERSONAL=3).
   e- classes **Check** and **CreditCard** implement interface ***Payable*** by implementing the method ***isAuthorized()*** as follows:
      - A **Check** payment is authorized if either the type of the check is CASHIER or if the amount of the payment is less than or equal to the ***accountBalance,*** otherwise it is not authorized.
      - A **CreditCard** payment is authorized if its ***expiryDate*** is less than or equal to the current date.

2- Create a Driver class that does the following:
   - Creates an ArrayList of type **CustomerPayment** and adds different types of payments to it (Cash, Check, CreditCard). Check or CreditCard payments should be checked if they are authorized before adding them to the list. If they are not authorized, they should not be added to the list. Make sure to deduct the amount of authorized Check payments (CERTIFIED and PERSONAL only) from the ***accountBalance*** before adding them to the list (use the method ***deductAmountFromBalance()***).

   - Sorts the ArrayList created (use **Collections.sort**) in ***descending*** order based on the calculated payment. After sorting you should print the list to the screen using the ***printPaymentInfo()*** method.

Your all field constructors should use EXACTLY the following format and order:

CustomerPayment (customerName, customerId, amount)

Cash (customerName, customerId, amount, discountRate)

Check (customerName, customerId, amount, accountNumber, accountBalance, type)

CreditCard (customerName, customerId, amount, chargingFee, expiryDate)

Sample Run:

Assume you *try* to add the following payments to your list:

```
new Check("Rana",7777,400,1111,350,Check.PERSONAL);
new Cash("Ahmad",4444,150,5.0);
new Check("Suha",5555,100,1111,200,Check.CASHIER);
new Check("Rania",7777,600.0,1111,750,Check.CERTIFIED);
new CreditCard("Randa",9999,170,20, new Date(124,05,03));
new CreditCard("Hani",6666,150,10, new Date(120,06,07));
```

Then the output of your program should be very similar to the following:

```
Check [accountNumber=1111, accountBalance=150.0, type=2, customerName=Rania,
       customerId=7777, amount=600.0] Payment = 600.0

CreditCard [chargingFee=20.0, expiryDate=Mon Jun 03 00:00:00 IDT 2024,
       customerName=Randa, customerId=9999, amount=170.0] Payment = 190.0

Cash [discountRate=5.0, customerName=Ahmad, customerId=4444, amount=150.0]
       Payment = 142.5

Check [accountNumber=1111, accountBalance=200.0, type=1, customerName=Suha,
       customerId=5555, amount=100.0] Payment = 100.0
```

IMPORTANT: What you need to turn in:

1- Your project folder (containing all your project files (*.java files) should be compressed (.rar) and saved as ***proj_youridnumber_yourLabsectionnumber.rar*** (for example if your student id number is 1221234 and your lab **section** is section 9 then the project folder should be called ***proj_1221234_s9.rar***). Turn in your project by ***replying to the course coordinator's message*** on itc under the lab meta section and attaching your project .rar file (***proj_youridnumber_yourLabsection.rar***).

2- You must include your full name, student id number, and lab section number in a comment at the beginning of each of your .java code files.

Late Projects (even one minute late) will NOT be accepted for any reason.
Project Files that are wrong (e.g. sending wrong files or .class files instead of .java files) or that are compressed incorrectly

or cannot be opened for any reason will not be graded and will receive a zero grade.

Any form of cheating or using unauthorized help resources (e.g. other students, chatGPT, …) will not be tolerated and will be penalized according to university regulations.