CS251: Intro to Software Engineering
Homework 2 (4 marks)
Due Date 26 April 2021 @ 11:00 pm (submit on Blackboard)

TA Name: ……………………………………… Mark: ……..

CS251 – Introduction to Software Engineering, 2021
Each student fills this form for his program and gives it to TA

The questions to answer about each program are included in the following form. **Print and fill this form and bring with you to the discussion.**

Student name: همام رامي مطيع عطار……………………**ID:**……**20160325**……**. Group: 3ALL_1**

Which Program (Idea for task 2) did you choose?

 Task 1 & half of Task 3.

Which of the following Java / OOP features did you use in your program?

I used inheritance (extend in Java)

I used Polymorphism (@OVERRIDE in java), and I used abstraction concept.

1. How many classes did you create and their names?

   Three (3) classes. Account, Normal Account, Special Account.

2. How many different access specifiers did you use and their names?

   #3 , public , private , protected

3. How many Java coding style rules did you use and which ones?

   1-Function (methods) Names , all function reflect the action does ,ex: deposit, withdraw , informBalance , addElement .

   2- variable and object name , I used Camel case on variable names and functions ex: accountArrayList, accountNumber,getName ,setBalance,

   3- Indentation , all my code respect the Indentation   .

   4- Complex Expressions , Avoid complex expressions when I loop on array list or when I retrieve objects from array list .

4. How many Javadoc tags did you use and which ones?

   1-@param

   2-@return

………

5. Did you use inheritance? When and why?

   I used it , that I have two type of account thy have same functionality , the different is in function withdraw and share same other functions , so instead of do two class and each class contain all functions , I created one super class named Account contain all shared functions then inherent the two type accounts with their specific functions , if client want to add new type of account I just will inherent form super class Account to inherent the general function of account .

   ………

6. Did you use method overriding? When and why?

   Yes sure , in function withdraw . as I mentioned above t, withdraw has difference restriction in difference accounts but they share general functions . so will be super class for general functions and override the withdraw that it will be different by different account type .

   ………

7. Did you use method composition? When and why?

   No I did not use .

   ………

8. Did you use method polymorphism? When and why?

   Yes , with classes Account , normal account and special account , that because here is method withdraw has different way to apply .

Draw in the space below a simple UML class diagram that shows your main classes, their attributes and operations and their interactions with each other.

Bank

- name :String
- address :String
- phone:String
-clientArrayList:Client [ ]
-accountArrayList:Account [ ]

+ setName (Name :String):void
+ setPhone (number:String):void
+ setAddress (Add : String):void
+ getName (Name :String):String
+ getPhone ():String
+ getAddress ():String
+toString() :String
+addElement(Client, Account):void
+diposit(int accountNumber,float amount):int
+withDraw(int accountNumber,float amount):int
+informBalance(int accountNumber):float
+disPlay():void

1   1

1..*   1..*

Client

- name :String

- address :String

- phone:String
- nationalID:String
- account : Account

+ Client(person :String , address:String
, phoneNumber:String , nationalID:String
,account : Account)
+ setName (Name :String):void
+ setPhone (number:String):void
+ setAddress (Add : String):void
+setNationalID(Id: String) :void
+setAccount (account: Account):void
+ getName (Name :String):String
+ getPhone ():String
+ getAddress ():String
+getNationalID() :String
+getAccount() :String
+toString() :String

1   1

Account

-balance:float

accountNumber :int

Validate

Account(balance :float ,
accountNumber :int)

getBalance():float

setBalance(balance :float) :void

getAccountNumber() :int

setAccountNumber(accountNumber: int):void

deposit(amount :float):boolean

withDraw(amount :float) :boolean

CommercialClient

Student Number

Average Mark

+ setCommercialID(Id : String) : void

+toString() :String

+getCommercialID() : String
CommercialClient(company :String , String address:String
, phoneNumber:String , companyID:String ,account :Account)

NormalAccount

+ field: type

+NormalAccount(balance :float
, accountNumber :int)

+ withDraw(amount :float) :boolean

SpecialAccount

+ field: type

+SpecialAccount(balance :float
, accountNumber :int)

+ withDraw(amount :float) :boolean

4