**Assignment#2, CS-2365, OOP**

**Consider the following Assignment Question. It consists of 18 parts, which are divided among groups of three students. For your portion of parts, coordinate with your group members to create one single solution. You have to submit the java file of your project (developed in NetBeans 8.1 IDE). Name the “java” file using the last names of all your group mates plus your group number. The java file is then compressed with your Netbeans project folder using winrar utility. Your java file must contain description of the parts of your assignment question in the comment section of your java code. Upload your compressed file on the black board. Hand-in: 04/04/2019, Hand-out: 03/07/2019. Note that in the lecture session instructor has discussed the creation of BankAccount class. However, the instructor has not yet taught the topics of GUI components’ creation. Therefore, to help students, instructor has uploaded an assignment2 handout on the blackboard. Instructor would also teach GUI topics in the class after the spring break. If anyone faces any problems in understanding the details of his/her tasks, kindly email the problem to the instructor.**

**Goals Of Assignement: To familiarize with inheritance and fundamental GUI components like PushButtons, TextBoxes and Radio Buttons (obligatory implementation of above 3 GUI components)**

**Total Marks: 10.0**

**Question. Create a Account class (fields private: String strName (TTU1, TTU2,..), int id(100, 101,…), char accountType i.e ‘C’ for current &‘S’ for saving accounts, double balance (hard-coded values). Provide get & set methods and parameterize & non-parameterize constructors. Also provide withdaw(double amount) & deposit(double amount) methods and displayBalance( ) and displayAll( ) method. Consult lecture#9 for this Assignment.**

**//displayAll( ) prints all instance variables**

**Create a SavingsAccount class inheriting from Account class. SavingsAccount class (Instance Variables: private: double rate ; Methods: public addInterest(…), deductLoan(…), parameterize constructor)**

***Note for simplicity, assume that SavingAccount class is managing both Saving (i.e. accountType =’s’) and Current ( i.e. accountType = ‘c’) accounHolders.***

**The first line of derived class constructor should invoke the base class constructor using super and then provide code to initialize the local instance variables.. You can hard-code the values also.**

**Now create a class UseSavingBankAccount class which incorporates GUI components. Create 50 array of objects for SavingAccount class using parameterize constructors.. Use a ‘for’ loop to create array of objects. Now provide radio button to perform following operations .**

**G1 (Guatum, Utkrist Dylan) G9 (Precious, Bharat, Sabin), G16(Thomas, Faus, Rahil)**

**Provide four radio Buttons (title same as operations in the question), three labels (Max/Min, Amount, AccountType), three text fields (to input ‘min’ or ‘max’, to input amount, and to input accountType) and a “submit” push button for processing the following two operations**

**(1)Find out and display the account id(s) & name(s) with minimum and maximum balance amount using JOptionPane. Provide methods findAccNameAndIDWithMinBalance(SavingAccount[]), and findAccNameAndIDWithMaxBalance(SavingAccount[]);**

**(2)Deposit amount in the lowest balance account. Hint: find accountID using findAccNameAndIDWithMinBalance(SavingAccount[]) and provide method Deposit(int iID, double Amount) which calls Deposit(double amount)**

**(3) Display balance of all current/saving AccountHolders using JOptionPane, provide method DisplayBal(char accountType) (3b) Exit Option**

RBOption: Find AcctID with Min/Max Balance

RBOption: Deposit Amount in Lowest Bal Account

RBOption: Display balance Using AccountType

TBLabel

TextBox:Inp ‘Max’ or ‘Min’

TextBox:Inp Amount

TBLabel

TBLabel

TextBox:Inp AccountType

Submit Buttonn

**G2 (Roshan, Michael, Liam) G10(Katlyn, Nicholas, Eduardo) G17(Isaak, William, Nathanael)**

**Provide three radio Buttons (title same as operations in the question), two labels(AccType, Name), two text fields (to input AccType and to input Name) and a “submit” push Button for the following three operations**

**(3) Display list of all BankAccounts with all fields (based upon accountType) using JOptionPane sorted by name. Provide a method sortByName(SavingAccount[], accountType).**

**(4) Provide a method searchbyName(…..) and display all the information of account holders using JOptionPane if “name” found. Note initially the “TextBox for inputting name” is disabled. But if we select this radio button then the “TextBox for inputting name” is enabled.). Provide a method searchByName(SavingAccount[] ……., String Name)**

**(4b) Exit Option**

RBOption: Disp BA list sorted by Name

RBOption: Search Account by Name

TBLabel

TextBox: Inp AccType

TextBox: Inp Name

TBLabel

Submit Buttonn

RBOption: Exit

**G3 (Pranjal, Joey, Abbey) G11(Mushfique, Eric, Edmond),G18(Haroon, Ivan Feebi)**

**Provide four radio Buttons (title same as operations in the question), three labels(Name, Name, and AccountType), three text fields (to input ‘max’ or ‘min’, to input Amount, and to input AccType) and a “submit” push Button for the following four operations**

**(5) Provide a method searchBalancebyName(….) of an account holder and display the result by JOptionPane. Note initially the “text box to input Name” is disabled. But if we select this radio button then the “text box” is enabled. searchBalanceByName(SavingAccount[] ……, String name)**

**(6) Add interest to the account holder’s balance by Name. addInterest method should call deposit(…).Note initially the “text boxes for inputting Name”and rate are disabled. But if we select this radio button then the “text boxes” are enabled. Provide a method:addInterestByName(SavingAccount[ ], String name, double rate)**

**(7) Display all information about account holders by accountType using JOptionPane.** **Provide a method: displayAllByAccountType(SavingAccount[ ], accountType);**

**(7b) Exit**

**(Note initially the “text boxes for inputting Name, Name and interest rate , and accountType” are disabled. But if we select this radio button then the “text boxes” are enabled.).**

RBOption: Search Balance by Name

RBOption: Add Interest to account Holder by Name

RBOption: DisplayAll Using AccountType

TBLabel

TextBox:Inp Name

TextBox1 :Inp Name, TB2: Inp rate

TBLabel

TBLabel

TextBox:Inp AccountType

Submit Buttonn

**G4 (Collin, Grant, Andrew) G12(Marie, Leslye, Trisha), G19(Anil, Santosh, Blaine) Provide four radio Buttons (title same as operations in the question), three labels (id, id, accountType), three text fields (to input id, again to input id and to input accountType) and a “submit” push Button for the following four operations**

**(8) Search the balance of an account holder by id and display balance by JOptionPane. Provide a method searchBalanceById(SavingAccount[] ….., int …..)**

**(9) Add an interest to the account holder by id. Provide method addInterestById(SavingAccount[] ….., int iID…., double rate) method should call deposit(…). Note addInterestById(…..) returns amount to deposit.**

**(10) Display all information about account holders by accountType using JOptionPane. Provide a method displayAllByAccountType (SavingAccount[ ], char accountType)**

**(10b) Exit**

**(Note initially the “Text boxes” are disabled for all above cases. But if we select the radio button then the “TextBoxes” are enabled.)**

RBOption: Search Balance by ID

RBOption: Add interest by id

RBOption: DisplayAll Using AccountType

TBLabel

TextBox:Inp ID

TextBox1:Inp ID, TextBox2: Inp rate

TBLabel

TBLabel

TextBox:Inp AccountType

Submit Buttonn

**G5 (Safwan, Alex, Tyler), G13(Anish, Demetrius, Domingo) G20(Deepen, Tecpal, Rick)**

**Provide three radio Buttons (title same as operations in the question), three labels(AccType, accountType and Amount), three text fields (to input AccType and to input amount and accountType) and a “submit” push Button for the following two operations**

**(11) Display all AccountHolders’ information by accountType sorted by balance in ascending order using JOptionPane. Provide Method: displayAllByAccountType(SavingAccount[], char); sortByBalance(SavingAccount[])**

**(12) withdraw loan amount from all account holders’ balance,. based upon accountType.. Provide a Method deductLoanByAccountType(SavingAccount[], char, double amount) should call withdraw(…).**

RBOption: DispAll by accountType

RBOption: WithDraw Loan by AccountType

TBLabel

TextBox: Inp AccType

TextBox1: Inp AccountTpe, TextBox2: Inp Amount

TBLabel

Submit Buttonn

RBOption: Exit

**G6 (Ronak, Saransh, Mohak) G14(Colyn, Chen, Jan) G22((Ihua, Bijay, Steven) Provide four radio Buttons (title same as operations in the question), two labels(ID, Name), two text fields (to input ID and to input Name) and a “submit” push Button for the following two operations**

**(13) Delete an account by id. Provide a method deleteAnAccountById(SavingAccount[], int)**

**(14) Delete an account by Name. Provide a method deleteAnAccountByName(SavingAccount[ ], String)**

**(15) Display all account holder’s information Sorted by name; Provide method sortByName(SavingAccount[ ]) and then call displayAll() (15b): Exit**

RBOption: Delete an account id

RBOption: Delete an account by Name

RBOption: Display all accountholder’s inform. Sorted by Name

TBLabel

TextBox:Inp ID

TextBox:Inp Name

TBLabel

Submit Buttonn

**G7 (Jeffrey, Chigozie, Samuel) G8 (Aaja, Sydney, Trevor),G15(Sohan, Benjamin, Chang)**

**Provide four radio Buttons (title same as operations in the question), two labels(AccType, Name), two text fields (to input AccType and to input Name) and a “submit” push Button for the following two operations**

**(16) Display all saving account holder’s/current account holder’s information sorted by balance in descending order based upon AccountType. Provide sortByBalanceDes(SavingAccount[ ], char accountType) and then call displayAll() method**

**(17) Alter the names of all Saving AccountHolders by appending the String for example “Lubbock”. Provide the method alterNames(SavingAccount[], char accountType, String);**

**(18) convert all current account into saving account and addInterest to all current account holders. Provide method convertCurrentToSaving(SavingAccount[], char, double)**

RBOption: Display all sorted by Bal using accontType

RBOption: Alter names of all accountHolders by accountType

RBOption: Convert Current to Saving and add interest

TBLabel

TextBox:Inp accountType

TextBox:Inp altering String

TBLabel

TBLabel

TextBox:Inp rate

Submit Buttonn