**Project, CS-2365, OOP**

**Project is the extension of your Assignment2 Question. Assignment2 focused on “from scratch design”. Project aims to develop a menu-based application using your “old Assignment2” details and any one sub-question related to your group. Before starting your project, you have to act on the following guidelines:**

1. **Remove your R#s from your Assignment2 source files**
2. **You have a new added member in your group, let’s call him “GUEST” and the original members are “HOST”**
3. **Initially the “GUEST” would take permission from all his previous group mates to share his code through email. Once this permission is sought, GUEST would share his file with HOST and give briefing about coding details. As an evidence, guest has to attached details of email communications with his/her former group mates.**
4. **If permission is not sought due to any reasons, HOST members would try to develop a consensus to share the code with the GUEST. If not then the group would develop the project from scratch. If consensus achieved in favor of sharing then you have to transform your GUI into a menu driven design from your existing radio button oriented design in Assignment#2.**

**Project consists of only 2 parts, one you would select from the first 3 options of the sub-question associated with your group and one is the exit option. For both these options you would create menu-items which would launch a user-defined dialog box and add all the GUI components on the dialog box based upon your selected sub question. For exit option, dialog box would contain only an exit button. Also one question requires individual answers which students can email directly to the professor by specifying “Project Individual Question + LastName + FirstName” attached in a word file named as “Project CS2365 Indivual Question + LastName + FirstName”. For your project parts (1/2/3 & exit option), coordinate with your group members to create one single solution. You have to create Java file of your project and then compress java source file with your whole NetBeans project developed in NetBeans 8.1 IDE. You would use winrar utility for compression. Name the compressed file using the last names of all your group mates plus your group number. Your java file must contain description of the part of assignment#2 question (which you have selected) in the comment section of your java code along with last + first names of all your group members (No R# required). Upload your compressed file on the black board. Hand-in: 04/15/2019, Hand-out: 04/27/2019. Note you would get information about Account and SavingsAccount class from lect#9 whereas Lect#13 (Gui3) can guide you how to create Menus and Menu Items. To help students, instructor has uploaded a Menu Application (i.e. Sample Dialog Box Application- For Project). You can check SampleDialogBox code from that application. If anyone faces any problems in understanding the details of his/her tasks, kindly email the problem to the instructor.**

**Goals Of Assignment (pertaining to the course): To familiarize students with the creation of Menu, MenuItems, and user defined Dialog Boxes.**

**(obligatory implementation of above 3 GUI components)**

**Required Project Implementations (i.e. Menu & MenuItems and so on)**

**Group-Based Tasks**

1. **Create a GUI Application**
2. **Insert a Menu**
3. **Insert 2 MenuItems in the Menu**
4. **Create a Dialog Box Class Associated With Each MenuItem**
5. **Insert GUI Elements in Each DialogBox**
6. **PopUp A Specific DialogBox When MenuItem is Clicked**

**If Consensus is Achieved for Sharing the Code:**

**Individual Task (Only Technical Points Allowed)**

**Design (only 2 Page)**

1. **Discuss in Your Own Words the Current Design of the Application You Have Received (i.e Discuss About Classes, Methods & DataMembers & Their Utilization to Achieve Application’s Goals in your radio button based application)**
2. **Discuss in Your Own Words Design of the Application After Your Enhancements (i.e Discuss About Classes, Methods & DataMembers You Have Added & Their Utilization to Achieve Application’s Goals in your Menu Driven Application)**
3. **Draw Class Diagram for Each Class You Have**

**Discussion: (Only 1 Page)**

1. **Discuss how this approach of code reusability has helped you.**
2. **Discuss any other technique by which you can Achieve Better Results**
3. **Instead of Account and SavingsAccount Class, discuss any other inheritance based scenario which can be applied in this Course**
4. **In the context of 3, describe sub-parts of an Assignment#2 related question using your example of Base and Derive classes..**

**If Consensus Not Achieved or you have 3 member group**

**Individual Task (Only Technical Points Allowed)**

**Design (only 2 Pages)**

1. **Discuss in Your Own Words the MenuBased Design of your Assignment#2 Application. (i.e Discuss About Classes, Methods & DataMembers & Their Utilization to Achieve Application’s Goals in your radio button based application)**
2. **Discuss in Your Own Words Design of the Application After You have created your Menu baed Application (i.e Discuss About Classes, Methods & DataMembers You Have Added & Their Utilization to Achieve Application’s Goals in your Menu Driven Application)**
3. **Draw Class Diagram for Each Class You Have**

**Discussion: (Only 1 Page)**

1. **Using Bottom Up Approach, create a design of your application as discussed in Lect#6**
2. **Instead of Account and SavingsAccount Class, discuss any other inheritance based scenario which can be applied in this Course Assignment#2 & Project**
3. **In the context of 3, describe sub-parts of an Assignment#2 related question using your example of Base and Derive classes.**

**Total Marks: 19.0**

**Question. Create a Account class (fields private: Customer , 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 Customer class (Instance Variables: private: String strLName ; private String strFName Methods: public get Methods & parameterize constructor)**

**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 like Menus, MenuItems and Dialog Boxes. Create 50 array of objects for SavingAccount class and 50 array of objects of Customer class using parameterize constructors. Use a ‘for’ loop to create array of objects. Now provide a menu “Options”:**

**G5 (Safwan, Alex, Tyler, Haroon: should download the file created by G18 from blackboard), G13(Anish, Demetrius, Domingo, Ivan: should download the file created by G18 from blackboard), G20 (Deepen, Tecpal, Rick, Feebi: should download the file created by G18 from blackboard), G8 (Aaja, Sydney, Trevor: should download the file created by G18 from blackboard)**

**And 2 menuItems (title of one of the menu items is same as operations in one of the sub-question 5, 6, 7 & 7b and the other is “exit”). Each menuitem launches a dialog box (i.e. dlgbox1, dlgbox2, dlgbox3, dlgbox4). GUI components in each dialog box are (created programmatically, see sample application on the blackboard, and) described below:**

**Based upon the subquestion selected, choose the label and text fields:**

**labels (Name: dlgBox1, Name & rate:dlgBox2 , and AccountType:dlgBox3), four text fields (to input Name: dlgBox1, to input Name & rate:dlgBox2, and to input AccType:dlgBox3) and a “submit” push Button (in each dlgBox) for the following four operations:**

**(5) Provide a method searchBalancebyName(….) of a Customer and display the result by JOptionPane. searchBalanceByName(SavingAccount[] ……, String name)**

**(6) Add interest to the Customer’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 Customers by accountType using JOptionPane.** **Provide a method: displayAllByAccountType(SavingAccount[ ], accountType);**

**(7b) Exit**

**(Note no need to disable the “text boxes” in these questions. Consider that all “text boxes” are enabled).**

**===**

**G3 (Pranjal, Joey, Abbey, Marie: Should download the file of G12 from black board) G11(shifted Mushfique, Eric, Edmond, Leslye: Should download the file of G12 from black board) G1 (Guatum, Utkrist, Dylan, Trisha: Should download the file of G12 from black board)**

**And 2 menuItems (title of one of the menu items is same as operations in one of the sub-question 8, 9, 10 & 10b and the other is “exit”). Each menuitem launches a dialog box (i.e. dlgbox1, dlgbox2, dlgbox3, dlgbox4). GUI components in each dialog box are (created programmatically, see sample application on the blackboard, and) described below:**

**Based upon the sub question selected, choose the label and text fields:**

**four labels (id: dlgBox1, id & rate:dlgBox2 , and AccountType:dlgBox3), four text fields (to input id: dlgBox1, to input id & rate:dlgBox2, and to input AccType:dlgBox3) and a “submit” push Button (in each dlgBox) for the following four operations:**

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

**(9) Add an interest to the Customer 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 Customers by accountType using JOptionPane. Provide a method displayAllByAccountType (SavingAccount[ ], char accountType)**

**(10b) Exit**

**(Note no need to disable the “text boxes” in these questions. Consider that all “text boxes” are enabled.)**

**===**

**G4 (Collin, Grant, Andrew, Ronak: Should download file of G6 created by Ronak, Saranash and Mohak), G19 (Anil, Santosh, Blaine, Saransh: Should download file of G6 created by Ronak, Saranash and Mohak), G2 (Roshan, Michael, Liam: Mohak: Should download file of G6 created by Ronak, Saranash and Mohak)**

**And four menuItems (title of one of the menu items is same as operations in one of the sub-questions 13, 14, 15 & 15b and the other is “exit” ). Each menuitem launches a dialog box (i.e. dlgbox1, dlgbox2, dlgbox3, dlgbox4). GUI components in each dialog box are (created programmatically, see sample application on the blackboard and) described below:**

**Based upon the subquestion selected; choose the label and text fields:**

**four labels (id: dlgBox1, id & rate:dlgBox2 , and AccountType:dlgBox3), four text fields (to input id: dlgBox1, to input id & rate:dlgBox2, and to input AccType:dlgBox3) and a “submit” push Button (in each dlgBox) for the following four 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 Customer’s information Sorted by name; Provide method sortByName(SavingAccount[ ]) and then call displayAll() (15b): Exit**

**==**

**G17 (Isaak, William, Nathanael, Chang: should download file of G15 from bb) G10 (Katlyn, Nicholas, Eduardo, Benjamin: should download file of G15 from bb) G16 (Thomas, Faus, Rahil, Sohan: should download file of G15 from bb) G9 (Precious, Bharat, Sabin, Samuel: should download file of G7) G22 (Ihua, Bijay, Steven, Jeffrey: should download file of G7) G14 (Colyn, Chen, Jan, Chigozie: should download file of G7)**

**And four menuItems (title of one of the menu items is same as operations in one of the sub-questions 16, 17, 18 &18b and the other is “exit” ). Each menuitem launches a dialog box (i.e. dlgbox1, dlgbox2, dlgbox3, dlgbox4). GUI components in each dialog box are (created programmatically, see sample application on the blackboard and) described below:**

**Based upon the subquestion selected, choose the label and text fields:**

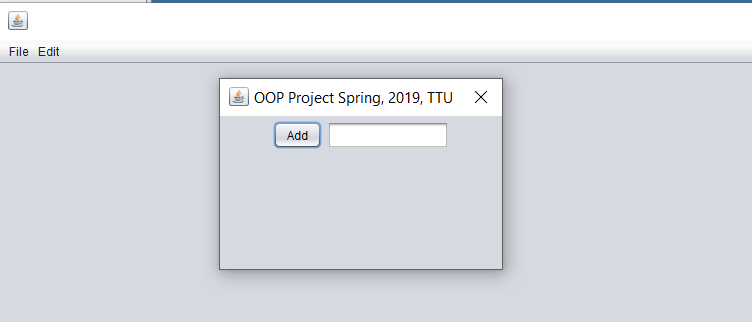
**two labels (AccountType: dlgBox1, String To Append:dlgBox2), two text fields (to input AccountType: dlgBox1, to input String To Append:dlgBox2) and a “submit” push Button (in each dlgBox) for the following four operations:**

**(16) Display all saving Customer’s/current Customer’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 Customer’s 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 Customers. Provide method convertCurrentToSaving(SavingAccount[], char, double)**

**(18b) exit**

****

**Output Of Sample Menu-Dialog Application**