**CSE 174 – Spring 2018**

**PROGRAM #13: 45 points – Due Tuesday, May 07, by 11:59 p.m.**

**Outcomes:**

* Write programs that use object oriented programming concepts.
* Write programs that use objects.
* Write programs that use ArrayList.
* Write applications using object oriented programming concepts.

**Scoring:**

* If you do not submit a .zip file containing your source code, your score will be zero.
* If you submit source code that does not compile, your score will be zero.
* If you submit source code without the correct class name (**Program13.java**) or the correct method names (see below), you will receive at most half credit for this assignment.
* Deductions will be made for not meeting the usual requirements:
  + Source code that is not formatted according to the usual guidelines, including commenting each method to explain its purpose. See program1 for an example of how we comment our methods.

|  |  |  |
| --- | --- | --- |
|  | **Full credit** | **Partial credit** |
| **Implement the Banking Application using ArrayList**  **(18 points)** | Your program implements the menu driven application program that simulates banking application as described. | Your program implements the banking application, but with some errors. |
| **Implement the methods createAccount(), display(), withdraw(), deposit(), displayAll(), removeAccount() ( 24 points)** | You implement these methods as described. | You implement these methods, but with some errors. |
| **Make sure the user does not enter bad values**  **(3 points)** | You make sure the user only enters a valid menu option from 1 up to 6 and other user inputs like amount, name,etc. | You check for bad user entry, but with some errors. |

**Assignment:**

1. Download BankAccount.java and Program13.java into a new folder called Program13.
2. Implement the menu driven Banking Application by completing the “STUBS” provided in **Program13.java** file using the **BankAccount** class provided in the folder. See the sample run below for the format you should match in your application.
3. Your solution should include completing the following methods which should be called when the user selects the corresponding menu options:

|  |
| --- |
| public static void createAccount(ArrayList<BankAccount> list) {  //creates an account  } |
| public static void display(ArrayList<BankAccount> list) {  //displays the specific account details  } |
| public static void displayAll(ArrayList<BankAccount> list){  // displays all the account details  } |
| public static void withdraw(ArrayList<BankAccount> list) {  // withdraws money from the specific object  } |
| public static void deposit(ArrayList<BankAccount> list) {  // deposits money into the specific object  } |
| public static void removeAccount(ArrayList<BankAccount> list){  // removes the specific account details  } |

1. Once your program is working well, use a loop to implement error checking, so that the user cannot enter illegal inputs and make sure that it terminates only on user’s choice 6.

**Sample run: (Text shown in red are user’s inputs)**

|  |
| --- |
| **\*\*\* Menu \*\*\***  **1. Create Account**  **2. Display**  **3. Display All**  **4. Deposit**  **5. Withdraw**  **6. Remove Account**  **7. Exit**  **Enter your choice: 5**  **No Accounts Created yet!!**  **\*\*\* Menu \*\*\***  **1. Create Account**  **2. Display**  **3. Display All**  **4. Deposit**  **5. Withdraw**  **6. Remove Account**  **7. Exit**  **Enter your choice: 1**  **\*\*Create New Account\*\***  **Enter name: John**  **Starting balance: 10000**  **Account Created!!**  **\*\*\* Menu \*\*\***  **1. Create Account**  **2. Display**  **3. Display All**  **4. Deposit**  **5. Withdraw**  **6. Remove Account**  **7. Exit**  **Enter your choice: 2**  **Enter Your name: John**  **\*\*Account Details\*\***  **Name: John Balance: 10000.0**  **\*\*\* Menu \*\*\***  **1. Create Account**  **2. Display**  **3. Display All**  **4. Deposit**  **5. Withdraw**  **6. Remove Account**  **7. Exit**  **Enter your choice: 3**  **Bank Account Details**  **Name: John Balance: 10000.0**  **\*\*\* Menu \*\*\***  **1. Create Account**  **2. Display**  **3. Display All**  **4. Deposit**  **5. Withdraw**  **6. Remove Account**  **7. Exit**  **Enter your choice: 4**  **\*\*Transaction - Deposit\*\***  **Enter Your Name: John**  **Enter amount to deposit: 1000**  **Name: John Balance: 11000.0**  **\*\*\* Menu \*\*\***  **1. Create Account**  **2. Display**  **3. Display All**  **4. Deposit**  **5. Withdraw**  **6. Remove Account**  **7. Exit**  **Enter your choice: 5**  **\*\*Transaction - Withdraw\*\***  **Enter Your Name: Johny**  **Name: Johny does not exist**  **\*\*\* Menu \*\*\***  **1. Create Account**  **2. Display**  **3. Display All**  **4. Deposit**  **5. Withdraw**  **6. Remove Account**  **7. Exit**  **Enter your choice: 1**  **\*\*Create New Account\*\***  **Enter name: Mathew**  **Starting balance: 2000**  **Account Created!!**  **\*\*\* Menu \*\*\***  **1. Create Account**  **2. Display**  **3. Display All**  **4. Deposit**  **5. Withdraw**  **6. Remove Account**  **7. Exit**  **Enter your choice: 3**  **Bank Account Details**  **Name: John Balance: 11000.0**  **Name: Mathew Balance: 2000.0**  **\*\*\* Menu \*\*\***  **1. Create Account**  **2. Display**  **3. Display All**  **4. Deposit**  **5. Withdraw**  **6. Remove Account**  **7. Exit**  **Enter your choice: 5**  **\*\*Transaction - Withdraw\*\***  **Enter Your Name: Mathew**  **Enter amount to withdraw: 100**  **Name: Mathew Balance: 1900.0**  **\*\*\* Menu \*\*\***  **1. Create Account**  **2. Display**  **3. Display All**  **4. Deposit**  **5. Withdraw**  **6. Remove Account**  **7. Exit**  **Enter your choice: 3**  **Bank Account Details**  **Name: John Balance: 11000.0**  **Name: Mathew Balance: 1900.0**  **\*\*\* Menu \*\*\***  **1. Create Account**  **2. Display**  **3. Display All**  **4. Deposit**  **5. Withdraw**  **6. Remove Account**  **7. Exit**  **Enter your choice: 6**  **Enter Your Name: John**  **Account Deleted!!**  **\*\*\* Menu \*\*\***  **1. Create Account**  **2. Display**  **3. Display All**  **4. Deposit**  **5. Withdraw**  **6. Remove Account**  **7. Exit**  **Enter your choice: 3**  **Bank Account Details**  **Name: Mathew Balance: 1900.0**  **\*\*\* Menu \*\*\***  **1. Create Account**  **2. Display**  **3. Display All**  **4. Deposit**  **5. Withdraw**  **6. Remove Account**  **7. Exit**  **Enter your choice: 7**  **Thanks for banking with us!!**  **>** |

**Advice:**

1. Your application uses an **ArrayList** called **list** that is capable of storing the **BankAccount** objects for any number of customers.
2. The **list.size()** method can be used in the loop to process(print/check for a specific account) all the BankAccount objects stored in the list.
3. At first, write the program without worrying about the user entering invalid numbers. Once the program is working correctly with valid numbers, then add the loop for handling bad numbers.