HW6: Bank Accounts: Static Members, Copy Constructors, and Overloading toString() and equals() Methods

Redo HW4 by adding several static members and adding overloaded toString() and equals() methods to each class as appropriate. In doing this, you will have to rewrite several other methods to have them make use of the implemented toString() and equals() methods.

Specifically, you should minimally do the following:

1. The **Bank** class: do not overload either the toString() or the equals() method.
   Add several static member variables and methods:
   totalAmountInSavingsAccts - sum total of balances in all Savings accounts
   totalAmountInCheckingAccts - sum total of balances in all Checking accounts
   totalAmountInCDAccts - total - sum total of balances in all CD accounts
   totalAmountInAllAccts - total - sum total of balances in all accounts
   Make sure to **provide appropriate methods** so as to allow for the **addition to, subtraction from, and reading of**, the current values of each of these static variables.
   Be sure to **print the values of all of these static variables when you print the database of accounts**.
2. The **Account** class: add a copy constructor and implements both the toString() and the equals() methods.
3. The **Depositor** class: add a copy constructor and implements both the toString() and the equals() methods.
4. The **Name** class: add a copy constructor and implements both the toString() and the equals() methods.
5. The **Check** class: add a copy constructor and implement the toString() method.
6. The **TransactionTicket** class: add a copy constructor and implement the toString() method.
7. The **TransactionReceipt** class: add a copy constructor and implement the toString() method.

Note: Make sure that when a method in an aggregate class returns a reference to a field object, it <u>returns a reference to a copy of the field object</u>.

Also, in addition to adding the static members, minimally rewrite the following methods so as to make use of the overloaded toString() methods:

printAccts() (make sure to include the printing of the static variables)
public static void printAccts(...);
public static void menu(...)
public static void balance(...);
public static void deposit(...);
public static void withdrawal(...);
public static void clearCheck(...);
public static void acctInfo(...);
public static void acctInfoHistory(...);
public static void newAcct(...);
public static void closeAcct(...);
public static void reopenAcct(...);
public static void deleteAcct(...);

As in previous assignments, make sure to use enough test cases so as to completely test program functionality.

The transaction methods in main() should "fill out" a TransactionTicket object, and then call the appropriate method within the Bank or Account class to carry out the requested transaction. The method should then print an appropriate transaction receipt using the toString() method of the TransactionReceipt class.

Submission Requirements:
Create a folder on Google Drive that will contain the following:
1. The source files (i.e., *.java files) for each of the implemented Classes:
pgmHW6.java
Bank.java; Account.java; Depositor.java, Name.java
Check.java; TransactionTicket.java; TransactionReceipt.java;
2. The text file containing the initial database of accounts (e.g., initAccounts.txt)
3. The test cases text file (e.g., myTestCases.txt)
4. The output text file which contains all of the required program output (e.g., pgmOutput.txt)
Then, make the folder shareable and send me a link to the folder.