```java
//Professor Ziegler
//HW7
//Jinglin Tan
//I am doing extra credit 2

import java.util.Scanner;   //needed to use Scanner
import java.io.*;           //needed to use PrintWriter

/*program 7 creates a small banking transaction system that can create/delete
 * accounts as well as perform deposits, withdrawals and balance inquiries
 */
public class pgm7{
    public static void main(String[] args) throws IOException{
        int num_accts;                          //number of accounts
        char choice;                            //choice of transaction
        final int MAX_NUM = 50;                 //maximum number of accounts
        int[] acctnum = new int[MAX_NUM];       //array for account numbers
        double[] balance = new double[MAX_NUM]; //array for balance of accounts

        File myfile1 = new File("c:/oldaccounts.txt");  //create file object
        Scanner input1 = new Scanner(myfile1);      //read old accounts

        File myfile2 = new File("c:/myinput.txt");  //create file object
        //Scanner input2 = new Scanner(System.in);
        Scanner input2 = new Scanner(myfile2);      //read my inputs

        //PrintWriter output = new PrintWriter(System.out);
        PrintWriter output = new PrintWriter("c:/myoutput.txt");  //output to file

        //get number of accounts from method readAccts
        num_accts = readAccts(acctnum, balance, MAX_NUM, input1, output);

        output.println("initial database of accounts and balances");
        printAccts(acctnum, balance, num_accts, output);    //print initial database

        do{
            menu(output);   //call menu() to print menu

            System.out.print("Please enter the selection: ");
            choice = input2.next().charAt(0);   //assign user input to choice

            switch (choice){
                case 'Q':
                    break;
                case 'W':
                    withdrawal(acctnum, balance, num_accts, input2, output);
                    break;
                case 'D':
                    deposit(acctnum, balance, num_accts, input2, output);
                    break;
                case 'N':
                    num_accts = newAcct(acctnum, balance, num_accts, input2, output);
                    break;
                case 'B':
                    balance(acctnum, balance, num_accts, input2, output);
                    break;
                case 'X':
                    num_accts = deletAcct(acctnum, balance, num_accts, input2, output);
                    break;
                default:
                    output.println("Error: " + choice + " is not in the menu");
```

```
                output.println();
                break;
            }
            output.flush();     //flush buffer after every transaction
        }while(choice != 'Q');

        output.println("Final contents of the account and balance arrays");
        printAccts(acctnum, balance, num_accts, output);  //print final database

        System.out.println("The program has completed");  //prompt program is finished
        input1.close();     //close input file
        input2.close();
        output.close();     //close output file
    }

    /* method readAccts()
     * Input: array of accounts, array of balances, number of maximum accounts,
     *        scanner object and printwriter object
     * Process: if number of accounts is less than maximum number of accounts and
     *          there is still something to read in the file, read old accounts
     *          from file
     *          count number of old accounts
     * Output:  return number of old accounts (num_accts)
     */
    public static int readAccts(int[] acctnum, double[] balance, int max_accts,
            Scanner input, PrintWriter output){
        int num_accts = 0;
        while(num_accts < max_accts && input.hasNext()){
            acctnum[num_accts] = input.nextInt();
            balance[num_accts] = input.nextDouble();
            num_accts++;
        }
        return num_accts;
    }

    /* method printAccts()
     * Input: array of accounts, array of balances, number of accounts
     *        and printwriter object
     * Process: loop through arrays of accounts and balances to print their values
     * Output:  print values of arrays of accounts and balance
     */
    public static void printAccts(int[] acctnum, double[] balance, int num_accts,
            PrintWriter output){
        for(int i = 0; i < num_accts; i++){
            output.printf("Account: %d", acctnum[i]);
            output.printf("   Balance: %.2f", balance[i]);
            output.println();
        }
        output.println();
        output.flush();
    }

    /* method menu()
     * Input: printwriter object
     * Process: print out menu of transactions on screen
     * Output:  print out menu of transactions on screen
     */
    public static void menu(PrintWriter output){
        System.out.println("Select one of the following:");
        System.out.println("W - Withdrawal");
        System.out.println("D - Deposit");
```

```
        System.out.println("N - New account");
        System.out.println("B - Balance");
        System.out.println("Q - Quit");
        System.out.println("X - Delete Account");
        System.out.println();
    }

    /* method findAcct()
     * Input: array of accounts, number of accounts, account number entered by user
     * Process: initialize integer variable index
     *          loop through array of accounts with index starting from 0
     *          if it finds an account number that matches the one entered by user,
     *           return the index of that account number
     *          if no match found, return -1
     * Output:  return index of matching account number
     *          or -1 if no match
     */
    public static int findAcct(int[] acctnum, int num_accts, int account){
        int index;
        for(index = 0; index < num_accts; index++){
            if(acctnum[index] == account)
                return index;
        }
        return -1;
    }

    /* method withdrawal()
     * Input: array of accounts, array of balances, number of accounts,
     *        scanner object and printwriter object
     * Process: prompt the user to enter an account number
     *          call method findAcct() to check if account exists
     *          if findAcct() returns -1, prints an error message
     *          else prompt the user to enter amount of withdrawal
     *           if the account balance has not enough fund, print an error message
     *           else if the amount is negative, print an error message
     *           else print a withdrawal confirm message and reduce amount from balance
     * Output:  print transaction type and account number entered
     *          print a message on screen basing on the validity of account number,
     *          amount entered and balance
     */
    public static void withdrawal(int[] acctnum, double[] balance, int num_accts,
            Scanner input, PrintWriter output){
        int account;
        int index;
        double amount;
        output.println("Transaction: withdrawal");
        System.out.print("Please enter the account number: ");
        account = input.nextInt();
        output.println("Account: " + account);
        index = findAcct(acctnum, num_accts, account);
        if(index == -1)
            output.println("Error: account " + account + " doesn't exist");
        else{
            System.out.print("Please enter the amount of withdrawal");
            amount = input.nextDouble();
            if(amount > balance[index]){
                output.printf("Error: account %d doesn't contain sufficient funds "
                        + "to withdraw %.2f", account, amount);
                output.println();
            }
            else if(amount < 0){
```

```
            output.printf("Error: the amount entered %.2f is negative", amount);
            output.println();
        }
        else{
            output.printf("Account %d has been withdrawed by %.2f", account, amount);
            output.println();
            balance[index] -= amount;
        }
    }
    output.println();
}

/* method deposit()
 * Input: array of accounts, array of balances, number of accounts,
 *        scanner object and printwriter object
 * Process: prompt the user to enter an account number
 *          call method findAcct() to check if account exists
 *          if findAcct() returns -1, prints an error message
 *          else prompt the user to enter amount of deposit
 *           if the amount is negative, print an error message
 *           else print a deposit confirm message and increase amount to balance
 * Output:  print transaction type and account number entered
 *          print a message on screen basing on the validity of account# and amount
 */
public static void deposit(int[] acctnum, double[] balance, int num_accts,
        Scanner input, PrintWriter output){
    int account;
    int index;
    double amount;
    output.println("Transaction: deposit");
    System.out.print("Please enter the account number: ");
    account = input. nextInt();
    output.println("Account: " + account);
    index = findAcct(acctnum, num_accts, account);
    if(index == -1)
        output.println("Error: account " + account + " doesn't exist");
    else{
        System.out.print("Please enter the amount of deposit: ");
        amount = input.nextDouble();
        if(amount < 0){
            output.printf("Error: the amount entered %.2f is negative", amount);
            output.println();
        }
        else{
            output.printf("Account %d has been deposited with %.2f", account, amount);
            output.println();
            balance[index] += amount;
        }
    }
    output.println();
}


/* method newAcct()
 * Input: array of accounts, array of balances, number of accounts,
 *        scanner object and printwriter object
 * Process: prompt the user to enter an account number
 *          call method findAcct() to check if account exists
 *          if findAcct() returns a non -1 number, which means the account already
 *           exists, then prints an error message
 *          else create an account by adding the account number one position after
```

```
 *           the last account in array, and add balance 0 one position after
 *           the last balance in array.
 *           print a new account creation confirm message
 *           increase the number of accounts by 1 and return it
 * Output:  print transaction type and account number entered
 *          prints an error if account already exists, otherwise a confirm message
 *          return the number of accounts (num_accts)
 */
public static int newAcct(int[] acctnum, double[] balance, int num_accts,
        Scanner input, PrintWriter output){
    int account;
    int index;
    output.println("Transaction: new account");
    System.out.print("Please enter the account number");
    account = input.nextInt();
    output.println("Account: " + account);
    index = findAcct(acctnum, num_accts, account);
    if(index != -1)
        output.println("Error: account " + account + " already exists");
    else{
        acctnum[num_accts] = account;
        balance[num_accts] = 0;
        output.println("Account " + account + " has been created with balance 0.00");
        num_accts++;
    }
    output.println();
    return num_accts;
}

/* method balance()
 * Input: array of accounts, array of balances, number of accounts,
 *        scanner object and printwriter object
 * Process: prompt the user to enter an account number
 *          call method findAcct() to check if account exists
 *          if findAcct() returns -1, prints an error message
 *          else print the balance of account
 * Output:  print transaction type and account number entered
 *          print balance of account if account exists
 *          or an error message if account doesn't exist
 */
public static void balance(int[] acctnum, double[] balance, int num_accts,
        Scanner input, PrintWriter output){
    int account;
    int index;
    output.println("Transaction: balance");
    System.out.print("Please enter the account number: ");
    account = input.nextInt();
    output.println("Account: " + account);
    index = findAcct(acctnum, num_accts, account);
    if(index == -1)
        output.println("Error: account " + account + " doesn't exist");
    else{
        output.printf("The balance of account %d is %.2f", account, balance[index]);
        output.println();
    }
    output.println();
}

/* method deletAcct()    (I am doing extra 2)
 * Input: array of accounts, array of balances, number of accounts,
 *        scanner object and printwriter object
```

```
     * Process: prompt the user to enter an account number
     *          call method findAcct() to check if account exists
     *          if findAcct() returns -1, print an error message
     *          else if the balance of the account is non-zero, print an error message
     *          else use a for loop to shift account numbers and balances that are after
     *           the index of the deleting account to left by one position, and assign 0
     *           to the original last account and last balance in the arrays
     *           print a confirm message of deletion
     *           reduce the number of accounts by 1 and return it
     * Output:  print transaction type and account number entered
     *          print a message basing on the validity of account for deletion
     *          return number of accounts (num_accts)
     */
    public static int deletAcct(int[] acctnum, double[] balance, int num_accts,
            Scanner input, PrintWriter output){
        int account;
        int index;
        output.println("Transaction: delete Account");
        System.out.print("Please enter the account number: ");
        account = input.nextInt();
        output.println("Account: " + account);
        index = findAcct(acctnum, num_accts, account);
        if(index == -1)
            output.println("Error: account " + account + " doesn't exist");
        else if(balance[index] != 0)
            output.println("Error: account " + account + " has a non-zero balance");
        else{
            for(int i = index; i < num_accts - 1; i++){
                acctnum[i] = acctnum[i + 1];
                balance[i] = balance[i + 1];
            }
            acctnum[num_accts - 1] = 0;
            balance[num_accts - 1] = 0;
            output.println("Account " + account + " has been deleted");
            num_accts--;
        }
        output.println();
        return num_accts;
    }
}
```