

```

/* Kymberlee Sables (n01364866)
   This program is an ATM Simulator that has users input an ID and select options.
   After the user exits, the ATM will be ready to use for the next user. This program
   implements loops, a class, etc. to accomplish its tasks.*/

//Imports
import java.util.Scanner;

//TestAccount Class
public class TestAccountSables {
    // Main Method
    public static void main(String[] args) {
        // Java Scanner
        Scanner input = new Scanner(System.in);

        // accountIDs (Step 1)
        int[] accountIDs = { 9801, 2345, 3025, 1024, 4096, 6021, 8192, 5768, 7113, 1497 };

        // Instance of class Account (Object)
        Account[] accounts = new Account[10];

        // Setting the IDs and initial balance values into the accounts (Step 2)
        for (int i = 0; i < accounts.length; i++) {
            accounts[i] = new Account();
            accounts[i].setId(accountIDs[i]);
            accounts[i].setBalance(Math.random() * 5000);
        }

        // infinite loop - enables the ATM to be ready for the next user after exiting
        // (Step 7)
        int option = 0;

        do {
            System.out.printf("*****Welcome! Welcome!*****\n");
            System.out.printf("*****\n\n");

            // Entering a user ID (Step 3 and Step 4)
            int userID;

            do {
                // user input
                System.out.printf("Please enter a correct ID: ");
                int id = input.nextInt();

                // calling linear search method
                userID = linearSearch(accountIDs, id);

                // prompting the user to re-enter an ID if invalid
                if (userID == -1)
                    System.out.printf("Invalid ID. Please try again.\n\n");
            } while (userID == -1);

            // display menu options (Step 5)
            do {
                System.out.printf("*****Main Menu*****\n");
                System.out.printf("\t\t1: check balance");
                System.out.printf("\n\t\t2: withdraw");
                System.out.printf("\n\t\t3: deposit");
                System.out.printf("\n\t\t4: exit");
                System.out.printf("\n\t\tEnter a choice: ");

                // choose menu options (Step 6)
                do {
                    option = input.nextInt();

                    if (option == 1) {
                        System.out.printf("*****\n");
                        System.out.printf("The balance is\t$ %.2f\n", accounts[userID].getBalance());
                    } else if (option == 2) {
                        System.out.printf("*****\n");
                        System.out.printf("Enter an amount to withdraw: ");
                        double withdrawAmount = input.nextDouble();

                        accounts[userID].withdraw(withdrawAmount);
                    } else if (option == 3) {
                        System.out.printf("*****\n");
                        System.out.printf("Enter an amount to deposit: ");
                        double depositAmount = input.nextDouble();
                    }
                } while (option != 4);
            } while (option != 4);
        } while (true);
    }
}

```

```

        accounts[userID].deposit(depositAmount);
    } else if (option == 4) {
        System.out.printf("*****\n");
        System.out.printf("*****Thank you! Have a nice day! Good bye!*****\n");
        System.out.printf("*****\n\n");
    } else if (option < 1 || option > 4) {
        System.out.printf("\t\tInvalid option. Please try again\n\t\tEnter a choice: ");
    }
    } while (option < 1 || option > 4);
} while (option != 4);

    } while (option == 4);

    input.close();
}

// Linear Search Method
public static int linearSearch(int[] accountIDs, int id) {
    for (int i = 0; i < accountIDs.length; i++) {
        if (accountIDs[i] == id)
            return i;
    }
    return (-1);
}

}

//Account Class
class Account {
    // variables (data field)
    private int id;
    private double balance;

    /*
     * default constructor - this constructor is provided by Java as a default
     * account when there are no parameters.
     */
    public Account() {
        // TODO Auto-generated constructor stub
    }

    /*
     * constructor (specified values) - this constructor gives specified values to
     * the data fields when there are parameters.
     */
    public Account(int i, double b) {
        id = i;
        balance = b;
    }

    // method(s) (Accessors) - these are methods that allow access to private data
    // fields.
    /*
     * getId - allows object "accounts" to get Id value.
     */
    public int getId() {
        return this.id;
    }

    /*
     * getBalance - allows object "accounts" to get balance value.
     */
    public double getBalance() {
        return this.balance;
    }

    // methods(s) (Mutators) - these are methods that allow private data fields to
    // be redefined.
    /*
     * setId - re-populates "accounts" with a set of values from an array.
     */
    public void setId(int id) {
        this.id = id;
    }

    /*
     * setBalance - sets the initial balances of "accounts" using the Math.random
     * class.

```

```

    */
    public void setBalance(double balance) {
        this.balance = balance;
    }

    /*
    * method (withdraw) - method that subtracts an amount from the initial balance.
    */
    public void withdraw(double withdrawAmount) {
        if (withdrawAmount > balance)
            System.out.println("The amount is too large, ignored!");
        else {
            balance -= withdrawAmount;
            System.out.printf("Take your cash. Your current balance is $%.2f\n", balance);
        }
    }

    /*
    * method (deposit) - method that adds an amount to the initial balance.
    */
    public void deposit(double depositAmount) {
        if (depositAmount < 0)
            System.out.println("The amount is negative, ignored");
        else {
            balance += depositAmount;
            System.out.printf("$ %.2f deposited. Your current balance is $ %.2f\n", depositAmount, balance);
        }
    }
}

```