UNIVERSITY OF ENGINEERING AND TECHNOLOGY, TAXILA


*SOFTWARE ENGINEERING*

# OOP-LAB 7

SUBMITTED TO: ENGR.SIDRA SHAFFI

SUBMITTED BY: MUHAMMAD ARSALAN

REG NO: 20-SE-56

COURSE: OOP-LAB

DATED: October 6, 2021

# TASK

## Part 1

File Account.java contains a definition for a simple bank account class with methods to withdraw, deposit, get the balance and account number, and return a String representation. Note that the constructor for this class creates a random account number. Save this class to your directory and study it to see how it works. Then modify it as follows:

1. Overload the constructor as follows:

• public Account (double initBal, String owner, long number) – initializes the balance, owner, and account number as specified

• public Account (double initBal, String owner) – initializes the balance and owner as specified; randomly generates the account number.

• public Account (String owner) – initializes the owner as specified; sets the initial balance to 0 and randomly generates the account number.

## Code:

```
package lab7;
import java.util.Scanner;
import java.util.Random;

class Account {

  Random rand=new Random();
private double balance;
private String name;
private long acctNum;
      public Account(String owner) {

            balance = 0;
            name = owner;
            acctNum = rand.nextInt(1000);
      }
      public Account(double initBal, String owner) {

            balance = initBal;
            name = owner;
            acctNum = rand.nextInt(1000);
      }
      public Account(double initBal, String owner, long number) {
            balance = initBal;
            name = owner;
            acctNum = number;
}

      public void withdraw(double amount) {

            if (balance >= amount)
            balance -= amount;

            else

                  System.out.println("Insufficient funds");
```

```
	}

	public void withdraw(double amount,double fee) {

		if (balance >= amount)
		balance -= (amount+fee);

		else

			System.out.println("Insufficient funds");
	}

	public void deposit(double amount) {

		if(amount<0){

			System.out.println("Invalid input. Negative Amount Can not be
deposited");
		}
	else
		balance += amount;
	}

	public double getBalance() {

		return balance;
	}

	public String toString() {

		return "Name:" + name + "\nAccount Number: " + acctNum + "\nBalance:
" + balance;
	} }


public class TestAccount {
	public static void main(String[] args) {

		String name;
		double balance;
		long acctNum;

		Account acct,acct1,acct2;

		Scanner scan = new Scanner(System.in);

		System.out.println("Enter account holder's first name");

		name = scan.next();

		acct = new Account(name);

		System.out.println("Account for " + name + ":");
		System.out.println(acct);
```

```
            System.out.println("\nEnter initial balance");

            balance = scan.nextDouble();

            acct1 = new Account(balance,name);
            System.out.println("Account for " + name + ":");
            System.out.println(acct1);
            System.out.println("\nEnter account number");

            acctNum = scan.nextLong();

            acct2 = new Account(balance,name,acctNum);
            System.out.println("Account for " + name + ":");
            System.out.println(acct2);
            System.out.print("\nDepositing 100 into account, balance is now ");

            acct2.deposit(100);
            System.out.println(acct2.getBalance());
            System.out.print("\nWithdrawing $25, balance is now ");

            acct2.withdraw(25);
            System.out.println(acct2.getBalance());
            System.out.print("\nWithdrawing $25 with $2 fee, balance is now ");

            acct2.withdraw(25,2);
            System.out.println(acct2.getBalance());
            System.out.println("\nBye!");
} }
```

```
package lab7;
import java.util.Scanner;
import java.util.Random;

class Account {

    Random rand=new Random();
private double balance;
private String name;
private long acctNum;
    public Account(String owner) {

        balance = 0;
        name = owner;
        acctNum = rand.nextInt(1000);
    }
    public Account(double initBal, String owner) {

        balance = initBal;
        name = owner;
        acctNum = rand.nextInt(1000);
    }
    public Account(double initBal, String owner, long number) {
        balance = initBal;
        name = owner;
        acctNum = number;
}
```

```java
    public void withdraw(double amount) {

        if (balance >= amount)
        balance -= amount;

        else

            System.out.println("Insufficient funds");
    }

    public void withdraw(double amount,double fee) {

        if (balance >= amount)
        balance -= (amount+fee);

        else

            System.out.println("Insufficient funds");
    }

    public void deposit(double amount) {

        if(amount<0){

            System.out.println("Invalid input. Negative Amount Can not be deposited");
        }
    else
        balance += amount;
```

```java
    }

    public double getBalance() {

        return balance;
    }

    public String toString() {

        return "Name:" + name + "\nAccount Number: " + acctNum + "\nBalance: " + balance;
    } }


public class TestAccount {
    public static void main(String[] args) {

        String name;
        double balance;
        long acctNum;

        Account acct,acct1,acct2;

        Scanner scan = new Scanner(System.in);

        System.out.println("Enter account holder's first name");

        name = scan.next();

        acct = new Account(name);
```

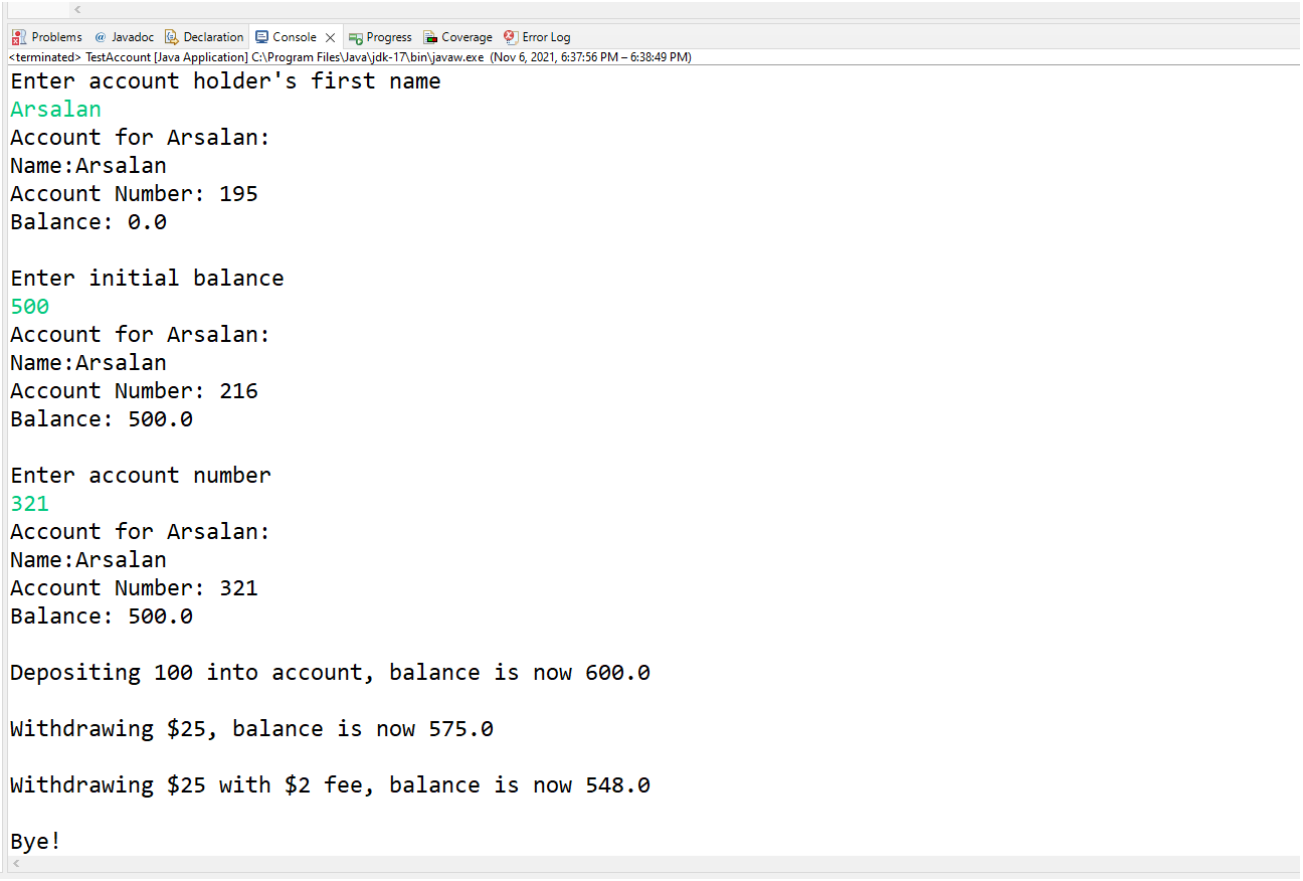
```
<terminated> TestAccount [Java Application] C:\Program Files\Java\jdk-17\bin\javaw.exe (Nov 6, 2021, 6:37:56 PM – 6:38:49 PM)
```

```
Main.java  *TestAccount.java  TestAccount1.java
 82          acct = new Account(name);
 83
 84          System.out.println("Account for " + name + ":");
 85          System.out.println(acct);
 86          System.out.println("\nEnter initial balance");
 87
 88          balance = scan.nextDouble();
 89
 90          acct1 = new Account(balance,name);
 91          System.out.println("Account for " + name + ":");
 92          System.out.println(acct1);
 93          System.out.println("\nEnter account number");
 94
 95          acctNum = scan.nextLong();
 96
 97          acct2 = new Account(balance,name,acctNum);
 98          System.out.println("Account for " + name + ":");
 99          System.out.println(acct2);
100          System.out.print("\nDepositing 100 into account, balance is now ");
101
102          acct2.deposit(100);
103          System.out.println(acct2.getBalance());
104          System.out.print("\nWithdrawing $25, balance is now ");
105
106          acct2.withdraw(25);
107          System.out.println(acct2.getBalance());
108          System.out.print("\nWithdrawing $25 with $2 fee, balance is now ");
109
110          acct2.withdraw(25,2);

Problems  Javadoc  Declaration  Console  Progress  Coverage  Error Log
<terminated> TestAccount [Java Application] C:\Program Files\Java\jdk-17\bin\javaw.exe  (Nov 6, 2021, 6:37:56 PM – 6:38:49 PM)
```

```
Main.java  *TestAccount.java  TestAccount1.java
 85          System.out.println(acct);
 86          System.out.println("\nEnter initial balance");
 87
 88          balance = scan.nextDouble();
 89
 90          acct1 = new Account(balance,name);
 91          System.out.println("Account for " + name + ":");
 92          System.out.println(acct1);
 93          System.out.println("\nEnter account number");
 94
 95          acctNum = scan.nextLong();
 96
 97          acct2 = new Account(balance,name,acctNum);
 98          System.out.println("Account for " + name + ":");
 99          System.out.println(acct2);
100          System.out.print("\nDepositing 100 into account, balance is now ");
101
102          acct2.deposit(100);
103          System.out.println(acct2.getBalance());
104          System.out.print("\nWithdrawing $25, balance is now ");
105
106          acct2.withdraw(25);
107          System.out.println(acct2.getBalance());
108          System.out.print("\nWithdrawing $25 with $2 fee, balance is now ");
109
110          acct2.withdraw(25,2);
111          System.out.println(acct2.getBalance());
112          System.out.println("\nBye!");
113 } }

Problems  Javadoc  Declaration  Console  Progress  Coverage  Error Log
<terminated> TestAccount [Java Application] C:\Program Files\Java\jdk-17\bin\javaw.exe  (Nov 6, 2021, 6:37:56 PM – 6:38:49 PM)
```




**PART-2**

**TestAccount1**

```
package lab7;
	//TestAccounts1
	// A simple program to test the numAccts method of the
	// Account class.
	import java.util.Scanner;
	public class TestAccount1 {
	public static void main(String[] args) {
	Account testAcct;
	Scanner scan = new Scanner(System.in);
	System.out.println("How many accounts would you like to create?");
	int num = scan.nextInt();
	for (int i=1; i<=num; i++) {
	testAcct = new Account(100, "Name" + i);
	System.out.println("\nCreated account " + testAcct);
	System.out.println("Now there are " + Account.numAccounts() + "
accounts");
	}
	}
	}
```

**PART-3**

**Write a test program that prompts for and reads in three names and creates an account with an initial balance of $100 for each. Print the three accounts, then close the first account and try to consolidate the second and third into a new account. Now print the accounts again, including the consolidated one if it was created.**

**CODE:**

```
package lab7;
import java.util.Random;
	import java.util.Scanner;
	import java.util.ArrayList;
	public class TestAccount2 {
	public static void main(String[] args) {
	Scanner input = new Scanner(System.in);
	System.out.println("How many accounts would you like to create?");
	int num = input.nextInt(); String[] names = {"First", "Second", "Third"};
	ArrayList<Account> owner = new ArrayList<Account>();
	for (int i=0; i<num; i++)
	{
	System.out.print("Account Names: "); names[i] = input.next();
owner.add(new Account(100, names[i]));
	}
	input.close();
	System.out.println(Account.getNumAccounts());
	for (int i=0; i<num; i++)
	{
	System.out.print("Account " + (i+1) + " is ");
	System.out.println(owner.get(i));
	}
	System.out.print("");
	owner.get(0).close();
	Account consolidatedAcc = Account.consolidate(owner.get(1), owner.get(2));
	System.out.print("New account is ");
	System.out.println(consolidatedAcc);
	System.out.println(Account.getNumAccounts());
	for (int i=0; i<num; i++)
	{
	System.out.print("Account " + (i+1) + " is ");
	System.out.println(owner.get(i));
	}
	System.out.print("New account is ");
	System.out.println(consolidatedAcc);
	}
	}
	 class Account
	{
	private static int numAccounts;
	private double balance;
	private String Name;
	private long accountNum;
	// Overloaded Constructors
	public Account(double initBal, String owner, long number)
	{
```

```
balance = initBal;
Name = owner;
accountNum = number;
numAccounts++;
}
public Account(double initialBal, String owner)
{
balance = initialBal;
Name = owner;
Random rand = new Random();
accountNum = rand.nextInt(1001);
numAccounts++;
}
public Account(String owner)
{
Name = owner;
balance = 0;
Random rand = new Random();
accountNum = rand.nextInt(1001);
numAccounts++;
}
public Account()
{
balance = 0;
Name = " ";
accountNum = 0;
numAccounts++;
}
// overloaded Method to withdraw Money
public void withdraw(double amount)
{
if (balance >= amount)
balance -= amount;
else
System.out.println("Insufficient funds");
}
public void withdraw(double amount, double fee)
{
if (amount <= 0 || fee<=0)
System.out.println("Error! Amount or fee can not be less than zero.");
else
{
double totalAmount = amount + fee;
if (balance >= totalAmount)
balance -= totalAmount;
else
System.out.println("Insufficient funds");
}
}
//Method to Deposit Money
public void deposit(double amount)
{
balance += amount;
}
```

```
        // Method to check balance
        public double getBalance()
        {
        return balance;
        }
        // Method to check number of accounts
        public static int getNumAccounts()
        {
        System.out.print("No. of Accounts created: ");
        return numAccounts;
        }
        // Method to close account
        public void close()
        {
        Name += "Closed";
        balance = 0;
        numAccounts--;
        }
        // Method to Consolidate Account
        public static Account consolidate(Account acct1, Account acct2)
        {
        if (acct1.Name.equals(acct2.Name) && acct1.accountNum != acct2.accountNum)
        {
        Account acct3 = new Account();
        acct3.Name = acct1.Name;
        acct3.balance = acct1.balance + acct2.balance;
        acct3.accountNum = acct2.accountNum + 1;
        acct1.close();
        acct2.close();
        return acct3;
        }
        else
        {
        System.out.println("Error! For consolidation Account owner must be the
same and must have two different accounts.");
        return null;
        }
        }
        public String toString()
        {
        return "Name:" + Name +
        "\nAccount Number: " + accountNum +
        "\nBalance: " + balance;
        }
        public static String numAccounts() {

        return null;
        }
```

```
    }
```

Main.java | TestAccount2.java

```
package lab7;
import java.util.Random;
    import java.util.Scanner;
    import java.util.ArrayList;
    public class TestAccount2 {
    public static void main(String[] args) {
    Scanner input = new Scanner(System.in);
    System.out.println("How many accounts would you like to create?");
    int num = input.nextInt(); String[] names = {"First", "Second", "Third"};
    ArrayList<Account> owner = new ArrayList<Account>();
    for (int i=0; i<num; i++)
    {
    System.out.print("Account Names: "); names[i] = input.next(); owner.add(new Account(100, names[i]));
    }
    input.close();
    System.out.println(Account.getNumAccounts());
    for (int i=0; i<num; i++)
    {
    System.out.print("Account " + (i+1) + " is ");
    System.out.println(owner.get(i));
    }
    System.out.print("");
    owner.get(0).close();
    Account consolidatedAcc = Account.consolidate(owner.get(1), owner.get(2));
    System.out.print("New account is ");
    System.out.println(consolidatedAcc);
```

Main.java | TestAccount2.java

```
27      System.out.println(Account.getNumAccounts());
28      for (int i=0; i<num; i++)
29      {
30      System.out.print("Account " + (i+1) + " is ");
31      System.out.println(owner.get(i));
32      }
33      System.out.print("New account is ");
34      System.out.println(consolidatedAcc);
35      }
36      }
37       class Account
38      {
39      private static int numAccounts;
40      private double balance;
41      private String Name;
42      private long accountNum;
43      // Overloaded Constructors
44      public Account(double initBal, String owner, long number)
45      {
46      balance = initBal;
47      Name = owner;
48      accountNum = number;
49      numAccounts++;
50      }
51      public Account(double initialBal, String owner)
52      {
```

Problems | Javadoc | Declaration | Console | Progress | Coverage | Error Log

<terminated> TestAccount2 [Java Application] C:\Program Files\Java\jdk-17\bin\javaw.exe (Nov 6, 2021, 7:50:25 PM – 7:52:17 PM)

```
Balance: 0.0
Account 3 is Name:ArsalanClosed
```

Main.java | TestAccount2.java

```
53      balance = initialBal;
54      Name = owner;
55      Random rand = new Random();
56      accountNum = rand.nextInt(1001);
57      numAccounts++;
58      }
59      public Account(String owner)
60      {
61      Name = owner;
62      balance = 0;
63      Random rand = new Random();
64      accountNum = rand.nextInt(1001);
65      numAccounts++;
66      }
67      public Account()
68      {
69      balance = 0;
70      Name = " ";
71      accountNum = 0;
72      numAccounts++;
73      }
74      // overloaded Method to withdraw Money
75      public void withdraw(double amount)
76      {
77      if (balance >= amount)
78      balance -= amount;
79      else
```

Problems | Javadoc | Declaration | Console | Progress | Coverage | Error Log

<terminated> TestAccount2 [Java Application] C:\Program Files\Java\jdk-17\bin\javaw.exe (Nov 6, 2021, 7:50:25 PM – 7:52:17 PM)

```
Balance: 0.0
```

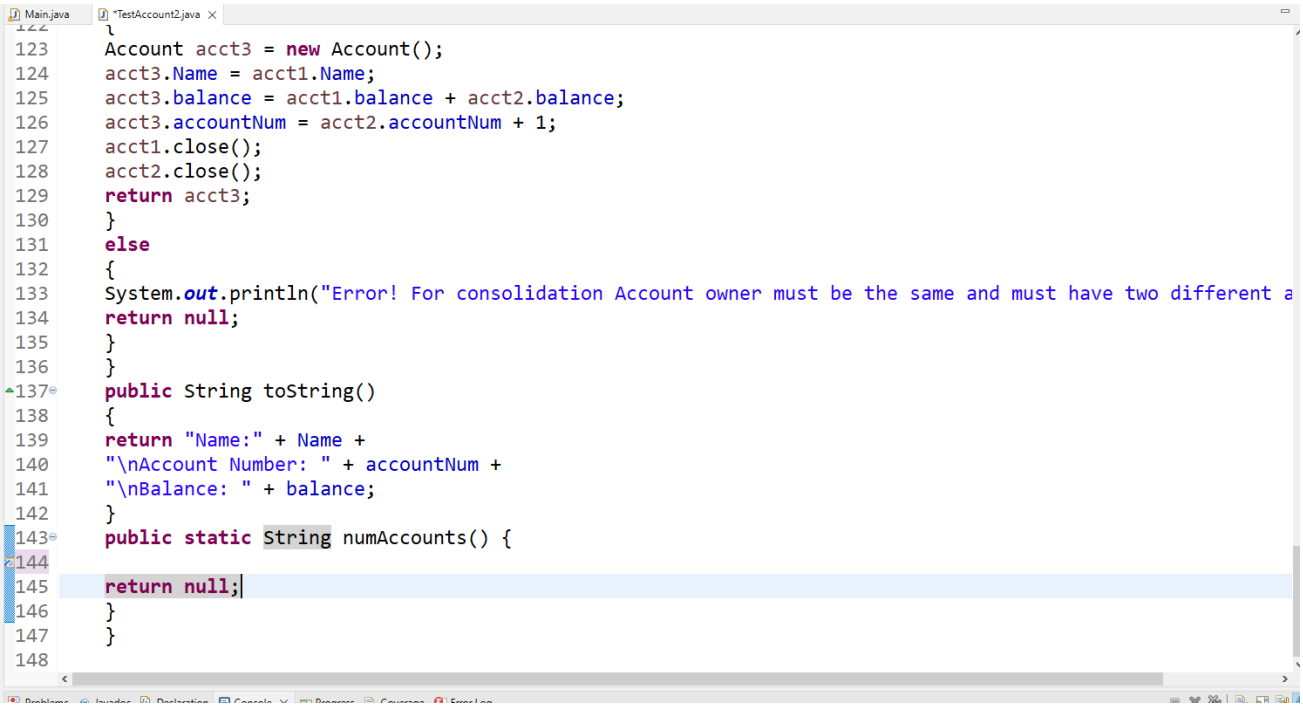
Main.java TestAccount2.java

```
77      if (balance >= amount)
78      balance -= amount;
79      else
80      System.out.println("Insufficient funds");
81      }
82      public void withdraw(double amount, double fee)
83      {
84      if (amount <= 0 || fee<=0)
85      System.out.println("Error! Amount or fee can not be less than zero.");
86      else
87      {
88      double totalAmount = amount + fee;
89      if (balance >= totalAmount)
90      balance -= totalAmount;
91      else
92      System.out.println("Insufficient funds");
93      }
94      }
95      //Method to Deposit Money
96      public void deposit(double amount)
97      {
98      balance += amount;
99      }
100     // Method to check balance
101     public double getBalance()
102     {
103     return balance;
```

Main.java TestAccount2.java

```
103     return balance;
104     }
105     // Method to check number of accounts
106     public static int getNumAccounts()
107     {
108     System.out.print("No. of Accounts created: ");
109     return numAccounts;
110     }
111     // Method to close account
112     public void close()
113     {
114     Name += "Closed";
115     balance = 0;
116     numAccounts--;
117     }
118     // Method to Consolidate Account
119     public static Account consolidate(Account acct1, Account acct2)
120     {
121     if (acct1.Name.equals(acct2.Name) && acct1.accountNum != acct2.accountNum)
122     {
123     Account acct3 = new Account();
124     acct3.Name = acct1.Name;
125     acct3.balance = acct1.balance + acct2.balance;
126     acct3.accountNum = acct2.accountNum + 1;
127     acct1.close();
128     acct2.close();
129     return acct3;
```

Problems Javadoc Declaration Console Progress Coverage Error Log


Main.java
*TestAccount2.java
123 Account acct3 = new Account();
124 acct3.Name = acct1.Name;
125 acct3.balance = acct1.balance + acct2.balance;
126 acct3.accountNum = acct2.accountNum + 1;
127 acct1.close();
128 acct2.close();
129 return acct3;
130 }
131 else
132 {
133 System.out.println("Error! For consolidation Account owner must be the same and must have two different a
134 return null;
135 }
136 }
137 public String toString()
138 {
139 return "Name:" + Name +
140 "\nAccount Number: " + accountNum +
141 "\nBalance: " + balance;
142 }
143 public static String numAccounts() {
144
145 return null;
146 }
147 }
148

Main.java | *TestAccount2.java

```
123     Account acct3 = new Account();
124     acct3.Name = acct1.Name;
```

Problems | Javadoc | Declaration | Console | Progress | Coverage | Error Log

<terminated> TestAccount2 [Java Application] C:\Program Files\Java\jdk-17\bin\javaw.exe (Nov 6, 2021, 7:50:25 PM – 7:52:17 PM)

```
How many accounts would you like to create?
3
Account Names: Bisma
Account Names: Arsalan
Account Names: Arsalan
No. of Accounts created: 3
Account 1 is Name:Bisma
Account Number: 676
Balance: 100.0
Account 2 is Name:Arsalan
Account Number: 173
Balance: 100.0
Account 3 is Name:Arsalan
Account Number: 706
Balance: 100.0
New account is Name:Arsalan
Account Number: 707
Balance: 200.0
No. of Accounts created: 1
Account 1 is Name:BismaClosed
Account Number: 676
Balance: 0.0
Account 2 is Name:ArsalanClosed
Account Number: 173
Balance: 0.0
Account 3 is Name:ArsalanClosed
Account Number: 706
Balance: 0.0
New account is Name:Arsalan
Account Number: 707
Balance: 200.0
```