Listing 1

```
 1 package edu.frontrange.csc240.exam1;
 2
 3 public class Account
 4 {
 5 private double balance;
 6 private int credits;
 7 private int debits;
 8 private double totalCredits;
 9 private double totalDebits;
10 public static final double CURRENT_RATE = 0.045;
11
12 public Account(double initialBalance)
13 {
14     if( initialBalance >= 0.0 )
15         balance = initialBalance;
16     else
17         showMessage("? Incorrect initial balance: $%.2f\n", initialBalance);
18 }
19
20 public Account(Account fromAccount)
21 {
22     balance = fromAccount.getBalance();
23 }
24
25 public void credit(double amount)
26 {
27     if( amount > 0.0 )
28     {
29         balance += amount;
30         totalCredits += amount;
31         credits++;
32     }
33     else
34         showMessage("? Invalid amount for a Credit: $%.2f\n", amount);
35 }
36
37 public void debit(double amount)
38 {
39     if( amount > 0.0 )
40         if( balance >= amount )
41         {
42             balance = balance - amount;
43             totalDebits = totalDebits + amount;
44             debits++;
45         }
46         else
47             showMessage("? Debit of $%.2f exceeds balance of $%.2f", amount,
48                     balance);
49     else
50         showMessage("? Invalid amount for a Debit: $%.2f\n", amount);
51 }
52
```

```
 53 public double getBalance()
 54 {
 55     return balance;
 56 }
 57
 58 public void displayValues()
 59 {
 60     displayValues("Account Values:");
 61 }
 62
 63 public void displayValues(String displayMessage)
 64 {
 65     showMessage(displayMessage);
 66     showMessage("   Balance is $%.2f\n", balance);
 67     showMessage("   %d credits totaling $%.2f\n", credits, totalCredits);
 68     showMessage("   %d debits totaling $%.2f\n", debits, totalDebits);
 69 }
 70
 71 public double futureValue(double years, double rate)
 72 {
 73     return balance * Math.pow(1.0 + rate, years);
 74 }
 75
 76 public double futureValue(int years, double rate)
 77 {
 78     double amount = balance;
 79     for( int count = 0; count < years; count++ )
 80         amount = amount + (amount * rate);
 81     return amount;
 82 }
 83
 84 public double futureValue(double years)
 85 {
 86     return balance * Math.pow(1.0 + CURRENT_RATE, years);
 87 }
 88
 89 private void showMessage(String message)
 90 {
 91     System.out.println(message);
 92 }
 93
 94 private void showMessage(String message, double firstValue)
 95 {
 96     System.out.printf(message, firstValue);
 97 }
 98
 99 private void showMessage(String message, double firstValue, double secondValue)
100 {
101     System.out.printf(message, firstValue, secondValue);
102 }
103
104 private void showMessage(String message, int firstValue, double secondValue)
105 {
106     System.out.printf(message, firstValue, secondValue);
107 }
108 }
```

Listing 2

```
 1 package edu.frontrange.csc240.exam1;
 2
 3 import java.util.Scanner;
 4 import java.util.Random;
 5
 6 public class AccountTester
 7 {
 8 Random randomNumbers = new Random();
 9 Scanner userInput = new Scanner(System.in);
10
11 public void testAccountClass()
12 {
13     System.out.println("\n\n*** Testing the Account Class ***");
14
15     System.out.printf("\nCurrent Interest Rate: %.2f%%", Account.CURRENT_RATE *
16             100.0);
17
18     System.out.println("\n\nCreating Account Objects");
19     Account accountOne = new Account(5000.0);
20     Account accountTwo = new Account(accountOne);
21     accountOne.displayValues("\nInitial Values for Account One:");
22     accountTwo.displayValues("\nInitial Values for Account Two:");
23
24     System.out.println("\n\nExercising Account Transactions");
25     accountOne.credit(0.0);
26     accountOne.debit(-6);
27     accountTwo.credit(-1.50);
28     accountTwo.debit(100.0);
29     accountTwo.debit(5000);
30     exerciseTransactions(accountOne, 100, 500.0);
31     exerciseTransactions(accountTwo, 250.0);
32     accountOne.displayValues("\n\nUpdated Values for Account One:");
33     accountTwo.displayValues("\nUpdated Values for Account Two:");
34
35     System.out.println("\n\nExercising Future Value Calculators");
36     System.out.printf("   Calculation 1: $%.2f\n", accountOne.futureValue(6.5,
37             0.05));
38     System.out.printf("   Calculation 2: $%.2f\n", accountOne.futureValue(15,
39             0.05));
40     System.out.printf("   Calculation 3: $%.2f\n", accountTwo.futureValue(6.5));
41     System.out.printf("   Calculation 4: $%.2f\n", accountTwo.futureValue(15));
42     System.out.println("\n\n*** Testing Complete ***\n");
43 }
44
45 private void exerciseTransactions(Account thisAccount, int testCount,
46         double limit)
47 {
48     int count = 0;
49     while( count < testCount )
50     {
51         if( (count % 3) != 0 )
52             thisAccount.credit(getDataValue(limit));
53         else
54             thisAccount.debit(getDataValue(limit));
55         count++;
```

```
 56     }
 57 }
 58
 59 private void exerciseTransactions(Account thisAccount, double limit)
 60 {
 61     char answer;
 62     displayMenu();
 63     do
 64     {
 65         answer = getUserCommand();
 66         switch( answer )
 67         {
 68         case 'D':
 69             thisAccount.debit(getUserValue(limit));
 70             break;
 71         case 'C':
 72             thisAccount.credit(getUserValue(limit));
 73             break;
 74         case 'V':
 75             thisAccount.displayValues();
 76             break;
 77         case 'M':
 78             displayMenu();
 79             break;
 80         }
 81     }
 82     while( answer != 'Q' );
 83 }
 84
 85 private double getDataValue(double maxValue)
 86 {
 87     return 1.0 + randomNumbers.nextDouble() * maxValue;
 88 }
 89
 90 private double getUserValue(double limit)
 91 {
 92     double thisValue;
 93     do
 94     {
 95         System.out.printf("Please enter a value (0.0 -  %.2f): ", limit);
 96         thisValue = userInput.nextDouble();
 97         if( !(thisValue >= 0.0 && thisValue <= limit) )
 98             System.out.printf("? Invalid input: %.2f\n", thisValue);
 99     }
100     while( thisValue < 0.0 || thisValue > limit );
101     return thisValue;
102 }
103
104 private void displayMenu()
105 {
106     System.out.println("\n\nPlease enter one of the following:");
107     System.out.println(" D - to test the debit method");
108     System.out.println(" C - to test the credit method");
109     System.out.println(" V - to display the values of the Account");
110     System.out.println(" M - to re-display this menu");
111     System.out.println(" Q - to exit this test\n");
112 }
```

```
113
114 private char getUserCommand()
115 {
116     char thisChar;
117     boolean goodChar;
118     do
119     {
120         System.out.print("Enter a command letter: ");
121         thisChar = userInput.next().toUpperCase().charAt(0);
122         goodChar = (thisChar == 'D' || thisChar == 'C' || thisChar == 'M' ||
123                 thisChar == 'Q' || thisChar == 'V');
124         if( !goodChar )
125             System.out.println("That is not a valid command letter");
126     }
127     while( goodChar == false );
128     return thisChar;
129 }
130 }
```

Listing 3

```
 1 package edu.frontrange.csc240.exam1;
 2
 3 public class ExamOneTest
 4 {
 5 public static void main( String args[] )
 6 {
 7     AccountTester testOne = new AccountTester();
 8     testOne.testAccountClass();
 9 }
10 }
```

Sample Output

```
*** Testing the Account Class ***

Current Interest Rate: 4.50%

Creating Account Objects

Initial Values for Account One:
   Balance is $5000.00
   0 credits totaling $0.00
   0 debits totaling $0.00

Initial Values for Account Two:
   Balance is $5000.00
   0 credits totaling $0.00
   0 debits totaling $0.00
```

```
Exercising Account Transactions
? Invalid amount for a Credit: $0.00
? Invalid amount for a Debit: $-6.00
? Invalid amount for a Credit: $-1.50
? Debit of $5000.00 exceeds balance of $4900.00

Please enter one of the following:
 D - to test the debit method
 C - to test the credit method
 V - to display the values of the Account
 M - to re-display this menu
 Q - to exit this test

Enter a command letter: D
Please enter a value (0.0 -  250.00): 117.50
Enter a command letter: d
Please enter a value (0.0 -  250.00): 25
Enter a command letter: c
Please enter a value (0.0 -  250.00): 125.75
Enter a command letter: V
Account Values:
   Balance is $4883.25
   1 credits totaling $125.75
   3 debits totaling $242.50
Enter a command letter: Q


Updated Values for Account One:
   Balance is $12567.09
   66 credits totaling $16098.65
   34 debits totaling $8531.56

Updated Values for Account Two:
   Balance is $4883.25
   1 credits totaling $125.75
   3 debits totaling $242.50


Exercising Future Value Calculators
   Calculation 1: $17256.99
   Calculation 2: $26126.07
   Calculation 3: $6500.77
   Calculation 4: $9450.47


*** Testing Complete ***
```