

Task 5.2.3 Practice Exercises - Working with Arrays of Objects

1. Assume that the application class below makes use of the Account class discussed previously

```
import java.util.*;
public class TestAccounts2
{
    public static void main(String args[])
    {
        String accountID, name;
        double balance;
        Scanner sc = new Scanner(System.in));

        /** Array used to store up to 10 accounts **/
        Account[] accounts = new Account[10];

        /** CODE for part a) goes here **/
        ...

        /** CODE for part c) goes here **/
        ...

        // print details for all accounts
        System.out.println("Account Details");

        /** Code for part b) goes here **/
        ...
    }
}
```

- a) Write code to populate the array with the Accounts containing the following details:

ID	Name	Balance
S5234	David	\$1000
S1239	Cliff	\$2000
S4236	Martin	\$3000
S8850	Jane	\$5000

Note that the array capacity will exceed the number of objects stored and thus need to consider how to manage working with this partially filled array in latter requirements.

- b) Write code which iterates through the objects that are currently stored in the array and prints their details to the screen. Your iteration loop should cease executing once it has gone past the last object in the array.

- c) Write code segments at the point following the comment for part C) in the sample code above which do each of the following:
- i) Iterate through the objects currently stored in the array and add 1% interest to any account with a balance > \$2000.
 - ii) Prompt user to enter an account ID, locate the Account object with that ID and process a withdrawal (getting the withdrawal amount from the user).

If the ID is not found or the withdrawal fails then an appropriate error message should be displayed to the user.

If the withdrawal is successful then display the new balance to the user.
 - iii) Prompt the user to enter the names of two Account holders to organise a transfer for, locate the Accounts with the specified names and proceed with a transfer from the first account to the second.

If either one or both of the accounts are not found, or the attempting transfer of funds fails, then a suitable error message should be displayed to the user.

If the transfer is successful then display the new balances for the two Accounts to the user.

Note that you should only use the one search loop to locate the two Accounts in question as this is a more efficient approach to the problem.

d. Account p,q;

```
for (int i=0;i<3; i++)
{
    if(accounts[i].getName().compareTo("Martin") == 0) p =
        accounts[i];
    else if(accounts[i].getName().compareTo("David") == 0) q =
        accounts[i];
}
p.transfer(q,100);
```

e. Replace the for loop with a do while loop so that the user may repeatedly print the account details by specifying the account ID. If no such ID exist print appropriate error message. After each iteration the user should be prompted whether to continue.

```
Enter Account ID:          s4236
Account details: Name = Martin      Balance = $3000
Continue (Y/N) ? Y

Enter Account ID:          s5234
Account details: Name = David      Balance = $1000
Continue (Y/N) ? Y

Enter Account ID:          s5444
No such account exist Continue
(Y/N) ? N

Bye for now.
```

(Hint: You need to use the getID() method)

1. What will be the output of the program below ?
diagrams.

Why ? Trace through with

```
class MyInt {
    private int val;

    public MyInt(int n)
    {
        val = n;
    }
    public void setVal(int n)
    {
        val = n;
    }
    public int getVal()
    {
        return val;
    }
}

public class TrySwap {
    public static void intSwap(int x, int y) {
        int temp = x;
        x = y;
        y = temp;
    }

    public static void refSwap(MyInt x, MyInt y) {
        MyInt temp = x;
        x = y;
        y = x;
    }

    public static void contentSwap(MyInt x, MyInt y) {
        int temp = x.getVal();
        x.setVal(y.getVal());
        y.setVal(temp);
    }

    public static void main (String[] args) {
        int u = 10;
        int v = 20;

        MyInt num1 = new MyInt(10);
        MyInt num2 = new MyInt(20);

        intSwap(u,v);
        System.out.println("u = " + u + " v = " + v );

        refSwap(num1, num2);
        System.out.println("num1 = " + num1.getVal() +
                           " num2 =" + num2.getVal() );

        contentSwap(num1, num2);
        System.out.println("num1 = " + num1.getVal() +
```

```
" num2 =" + num2.getVal() );
```

```
}  
}
```