Class Bank

```
1  //Bank class implemented using Arrays - Employee stuff has been removed for clarity
2  public class Bank
3  {
4      private String bankName;
5      private BankAccount[] accounts;          //declares array of object references
6
7      //default constructor
8      public Bank()
9      {
10         bankName = "Piggy Bank";
11         accounts = new BankAccount[10];      //creates Array object and limits it to 10
12         for (int i = 0; i < 10; i++)
13         {
14             accounts[i] = new BankAccount();        //fills each element in array with a refernce to a
   newly made bankAccount object
15         }
16
17         //alternate way of filling array using a temporary variable to refer to objects
18 //         for (int i = 0; i < 10; i++)
19 //         {
20 //             BankaAccount b = new BankAccount();
21 //             accounts[i] = b;   //fills each element in array with a refernce to the bankAccount obje
   ct that is also referred to by b
22 //         }
23     }
24
25     //this constructor requires that a reference to an already created array be passed as parameter
26     public Bank (String name, BankAccount[] premadeArray)
27     {
28         accounts = premadeArray;
29         bankName = name;
30     }
31
32     //this constructor takes a parameter for the number of accounts to create and creates an array obje
```

```
1  //Test class for Bank class implemented with arrays
2  public class TestArrayBank
3  {
4      public static void main (String[] args)
5      {
6          //create an array of of 5 BankAccounts for use in testing
7          BankAccount[] premadeBank = new BankAccount[5];
8          for (int i = 0; i < 5; i++)
9          {
10             premadeBank[i] = new BankAccount();
11         }
12
13         //Create a Bank object using each contsructor
14         Bank b1 = new Bank();
15         Bank b2 = new Bank("Liberty Bank", premadeBank);  //pass refernc
   e to array that was made above
16         Bank b3 = new Bank("Funky Bank", 15, 100);
17
18         b1.printCompanyReport();
19         b2.printCompanyReport();
20         b3.printCompanyReport();
21
22         //Add $500 to an account of your choice from each Bank
23         //these are each done a difference way - all are correct
24
25         //#1 using local variables for array and account references
26         BankAccount[] anArray;
27         anArray = b1.getAccounts();
28         BankAccount b = anArray[3];
29         b.deposit(500);
30
31         //#2 using local variables for array reference
32         BankAccount[] anotherArray;
33         anotherArray = b1.getAccounts();
34         anotherArray[0].deposit(500);
35
36         //#3 no local variables needed
37         b3.getAccounts()[2].deposit(500);
38
39
40         b1.printCompanyReport();
41         b2.printCompanyReport();
42         b3.printCompanyReport();
43
44         //Print the the total funds for each Bank.
45         System.out.println("Bank 1: " + b1.calcTotalFunds());
46         System.out.println("Bank 2: " + b2.calcTotalFunds());
47         System.out.println("Bank 3: " + b3.calcTotalFunds());
48
```

```
49          //Run the company report for each Bank.
50          b1.printCompanyReport();
51          b2.printCompanyReport();
52          b3.printCompanyReport();
53      }
54  }
55
```

```
32  ct of that size
33      //it loops through and creates BankAccount objects with the given starting balance and stores refer
    nces to them in the array field
34      public Bank(String name, int numOfAccounts, double startingBalance)
35      {
36          accounts = new BankAccount[numOfAccounts];
37          bankName = name;
38          for (int i = 0; i < accounts.length; i++)
39          {
40              accounts[i] = new BankAccount(startingBalance);
41          }
42      }
43
44      //Gets balance of each account, sums them, and returns total
45      public double calcTotalFunds()
46      {
47          double total = 0.0;
48          for (int i = 0; i < accounts.length; i++)
49          {
50              total += accounts[i].getBalance();
51          }
52          return total;
53      }
54
55      //subtracts a given fee from each account in the bank
56      public void chargeFees(double feeAmount)
57      {
58          for (int i = 0; i < accounts.length; i++)
59          {
60              accounts[i].withdraw(feeAmount);
61          }
62      }
63
64      //adds 5% interest to all accounts.
```

```
65      public void addInterest()
66      {
67          for (int i = 0; i < accounts.length; i++)
68          {
69              double interest = .05 * accounts[i].getBalance();
70              accounts[i].deposit(interest);
71          }
72      }
73
74      //accessor method which returns a refernce to the array of accounts
75      public BankAccount[] getAccounts()
76      {
77          return accounts;
78      }
79
80      //prints slips for all accounts.
81      public void printCompanyReport()
82      {
83          System.out.println("Bank Report:");
84          for (int i = 0; i < accounts.length; i++)
85          {
86              accounts[i].printSlip();
87          }
88      }
89  }
90
91
```