**Using the *Comparable* Interface**

Modify Question class (in the slides) to implement Comparable interface to compare two questions based on ComplexityLevel. Also write a driver class to test it, in other words, in the driver, create a few question objects and call compareTo to compare question objects.

**A Flexible Account Class**

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.

2. Overload the *withdraw* method with one that also takes a fee and deducts that fee from the account.

File *TestAccount.java* contains a simple program that exercises these methods. Save it to your directory, study it to see what it does, and use it to test your modified Account class.

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Account.java

//

// A bank account class with methods to deposit to, withdraw from,

// change the name on, and get a String representation

// of the account.

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

public class Account

{

private double balance;

private String name;

private long acctNum;

//-------------------------------------------------

//Constructor -- initializes balance, owner, and account number

//-------------------------------------------------

public Account(double initBal, String owner, long number)

{

balance = initBal;

name = owner;

acctNum = number;

}

//-------------------------------------------------

// Checks to see if balance is sufficient for withdrawal.

// If so, decrements balance by amount; if not, prints message.

//-------------------------------------------------

public void withdraw(double amount)

{

if (balance >= amount)

balance -= amount;

else

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

}

//-------------------------------------------------

// Adds deposit amount to balance.

//-------------------------------------------------

public void deposit(double amount)

{

balance += amount;

}

//-------------------------------------------------

// Returns balance.

//-------------------------------------------------

public double getBalance()

{

return balance;

}

//-------------------------------------------------

// Returns a string containing the name, account number, and balance.

//-------------------------------------------------

public String toString()

{

return "Name:" + name +

"\nAccount Number: " + acctNum +

"\nBalance: " + balance;

}

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// TestAccount.java

//

// A simple driver to test the overloaded methods of

// the Account class.

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

import java.util.Scanner;

public class TestAccount

{

public static void main(String[] args)

{

String name;

double balance;

long acctNum;

Account acct;

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();

acct = new Account(balance,name);

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

System.out.println(acct);

System.out.println("\nEnter account number");

acctNum = scan.nextLong();

acct = new Account(balance,name,acctNum);

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

System.out.println(acct);

System.out.print("\nDepositing 100 into account, balance is now ");

acct.deposit(100);

System.out.println(acct.getBalance());

System.out.print("\nWithdrawing $25, balance is now ");

acct.withdraw(25);

System.out.println(acct.getBalance());

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

acct.withdraw(25,2);

System.out.println(acct.getBalance());

System.out.println("\nBye!");

}

}

**Opening and Closing Accounts**

File *Account.java* (see previous exercise) 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 write the following additional code:

1. Suppose the bank wants to keep track of how many accounts exist.
2. Declare a private static integer variable numAccounts to hold this value. Like all instance and static variables, it will be initialized (to 0, since it’s an int) automatically.
3. Add code to the constructor to increment this variable every time an account is created.
4. Add a static method *getNumAccounts* that returns the total number of accounts. Think about why this method should be static - its information is not related to any particular account.
5. File *TestAccounts1.java* contains a simple program that creates the specified number of bank accounts then uses the getNumAccounts method to find how many accounts were created. Save it to your directory, then use it to test your modified Account class.
6. Add a method void close() to your Account class. This method should close the current account by appending “CLOSED” to the account name and setting the balance to 0. (The account number should remain unchanged.) Also decrement the total number of accounts.
7. Add a static method *Account consolidate(Account acct1, Account acct2)* to your Account class that creates a new account whose balance is the sum of the balances in acct1 and acct2 and closes acct1 and acct2. The new account should be returned. Two important rules of consolidation:

* Only accounts with the same name can be consolidated. The new account gets the name on the old accounts but a new account number.
* Two accounts with the same number cannot be consolidated. Otherwise this would be an easy way to double your money!

Check these conditions before creating the new account. If either condition fails, do not create the new account or close the old ones; print a useful message and return null.

1. 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.

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// TestAccounts1

// A simple program to test the numAccts method of the

// Account class.

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

import java.util.Scanner;

public class TestAccounts1

{

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");

}

}

}

**Transfering Funds**

File *Account.java* (see **A Flexible Account Class** exercise) contains a definition for a simple bank account class with methods to withdraw, deposit, get the balance and account number, and print a summary. Save it to your directory and study it to see how it works. Then write the following additional code:

1. Add a method *public void transfer(Account acct, double amount)* to the Account class that allows the user to transfer funds from one bank account to another. If *acct1* and *acct2* are Account objects, then the call *acct1.transfer(acct2,957.80)* should transfer $957.80 from acct1 to acct2. Be sure to clearly document which way the transfer goes!

2. Write a class TransferTest with a main method that creates two bank account objects account1 and account2 and then call to transfer 100$ from account1 to account2. Then prints a summary for each account.

3. Add a static method to the Account class that lets the user transfer money between two accounts without going through either account. You can (and should) call the method transfer just like the other one - you are overloading this method.

Your new method should take two Account objects and an amount and transfer the amount from the first account to the second account. The signature will look like this:

public static void transfer(Account acct1, Account acct2, double amount)

Modify your TransferTest class to use the static transfer instead of the instance version.