/45 Name: _____ Period: _____

APCS Problem Set 11: `abstract` Classes and `interfaces`

3.1 Motivating the Need for Abstract Classes: Enhancing a Subclass

Your job is to add a method to `SavingsAccount` that returns a `double` representing the balance in the account after '`y`' years have passed.[1] Here's the method signature:

```
public double getFutureBalance(int y)
```

As for implementation details, you have a choice:

a. write a loop that compounds the interest '`y`' times, or

b. take advantage of a compound interest formula involving an exponent.[2]

To test your method, add a `main()` to `SavingsAccount`, create an instance of `SavingsAccount`, and call `getFutureBalance()` on the object, making sure to print the results to the console. Here are test values you'll be asked to show:

Principal[a]	**APR**[b]	**Years**	**Ending Balance**
$725.00	3.75%	5	$871.52
$5,000.55	3.75%	8	$6,713.09
$25,000.18	3.75%	11	$37,481.10

[a] Starting balance.
[b] *Annual Percentage Rate*, or simply the interest rate.

Your program must print currencies neatly, rounding to the nearest penny and using commas to separate thousands. Using `System.out.printf()` instead of `System.out.println()` to print the values returned by `getFutureBalance()` makes all this quick and easy to do. `System.out.printf()` will even round to the nearest penny for you!

Teacher's Initials: _____ (10pts)

3.6 Book Questions

5. Litvin Ch. 12, #5: Create a new project called `PS11-Poem` and import `Poem.java` & `PoemDriver.java` from http://feromax.com/apcs/problemsets/PS11/downloads/Poem/. Afterward, implement the `Haiku` and `Limerick` classes, as described in the text.

Teacher's Initials: _____ (15pts)

[1] The interest is compounded annually.
[2] Look up `Math.pow()` in the Java API.

4.3 Polymorphism Works with `interfaces`, Too!

Just as subclasses with a common superclass can all be referred to as an object of the superclass' type (e.g., instances of `CheckingAccount` and `SavingsAccount` being handled as `BankAccounts`), classes with a common `interface` can all be referred to as objects of the `interface`'s type.

Your job is to write a new concrete class that `implements` `Actor`, much in the same way that `SoapOperaStar` currently does. Then, write a driver class that does the following:

- creates instances of `SoapOperaStar` and the class of your making,
- stores them in an `ArrayList`,
- and then retrieves them one at a time, calling `printQuote()` on each.

Teacher's Initials: _____ (10pts)

4.6 `implementing the` *`Comparable`* `interface`

If you look for the *`Comparable`* `interface` in the Java API, you'll see that (a) there are many classes that make use of it, and (b) it specifies only one method:

```
int compareTo(T o);
```

(Note: '`T`' stands for the *type* of object being compared to the current one.) Read at least the first sentence in the "method detail" section to understand what this method should return.

Your task: Go back to project `PS11-BankAccount` and change the `BankAccount` class so that it `implements` *`Comparable`*. (Yes, you can have an `abstract` class that implements an `interface`!) Use this class header line:

```
public abstract class BankAccount implements Comparable<BankAccount> {...}
```

You'll then notice that you're forced to implement the `compareTo()` method in each class `extending` `BankAccount`. That's not really true, though. If you write a complete `compareTo()` method in the superclass, `BankAccount`, writing separate versions in `CheckingAccount` and `SavingsAccount` is unnecessary.

Once you're finished adding `compareTo()` to `BankAccount`, add these lines to `BankAccountDriver`'s `main()`:

```
CheckingAccount c1 = new CheckingAccount("820-282-0216", 2165.09);
SavingsAccount s1 = new SavingsAccount("820-282-0216", 2165.11);

if ( c1.compareTo(s1) < 0 ) { //then c1 has less $$
    System.out.println("c1 has a lesser balance than s1.");
} else {  //compareTo() returns 0 or positive
    System.out.println("c1 has at least as much money as s1.");
}
```

Experiment with changing the account balances to see if the printed statements are accurate.

Teacher's Initials: _____ (10pts)