

Submission Instructions

For each of the assignments below, you must submit a zip file containing:

1. All source code, project files and any other files required to run the programs.
(Just zip up the entire project folder and submit that)
2. Screenshots of the program's output or error messages if the program doesn't work.

Concepts applied in these exercises

- Object Oriented Design
- Interfaces
- Classes
- Inheritance

Base Class

Write an **Account** class containing an instance variable called **balance** with an accessor called **getBalance** and mutators called **deposit** and **withdraw**. To allow members of the **Account** class to be sorted, implement the **java.lang.Comparable** interface to compare two accounts using **balance**.

Derived Classes

Create two subclasses **Account** called **SavingsAccount** and **CheckingAccount**.

SavingsAccount will add an instance variable called **interest** to store interest percentage and a mutator method for increasing its balance by applying the interest called **compound()**.

CheckingAccount will keep track of check numbers using an instance variable called **nextCheckNumber** starting with 1000. It will have a method called **orderChecks** that increments the current check number instance variable by 1000.

To facilitate displaying account information each of these accounts should override the **toString()** method from the **java.lang.Object** class.

Driver

All of your classes should include constructors that yield properly initialized accounts. Your driver should test your **SavingsAccount** and **CheckingAccount** classes by creating an **ArrayList** of **Account** and adding a number of **SavingsAccount** and **CheckingAccount** objects with various **balances**. All methods and constructors should be called at least once and your **ArrayList** should be passed as an argument to **java.util.Collections.sort()**.