Assignment 3: Portfolio Maps

(Last update: 2/26/2014) This assignment is designed to help you become more familiar with the Java Collections API. Create the following classes and add your own methods as necessary:

```
class Investment implements Comparable<Investment>{
   private String ticker;   // unique string identifier for stocks
   private int numShares;
   private double price;

   public boolean equals( Object rhs );
   public int compareTo( Investment rhs );
}

class Portfolio implements Comparable<Portfolio> {
   private String protfolioId;
   private double cashBalance;
   private List<Investment> holdings = new LinkedList<>();
}

// All the portfolios owned by a single customer
class Account extends Comparable<Account>, Iterable<Portfolio> {
   private String accountId;
   private List <Portfolio> portfolios;

   void addPortfolio( Portfolio );
   Iterator<Portfolio> iterator();
}
```

Let's assume an Account relates to a single customer, and we have lots of customers. Create an **AccountManager** class that contains a Map of Account objects, using each Account ID as the key:

```
class AccountManager {

   private Map<String,Account> accounts;        // Account ID is the key

   boolean readAccountsFile( String filename );
   /* Read a text file containg several accounts, store this information
      in the accounts map. Return True if successful. */

   List<Investment> getInvestmentList()
   /* Returns a list all stocks (investments), sorted in ascending
      order by ticker symbol.*/

   double getStockValuation( String accountId )
   /* Returns the sum value of all stock holdings in a
      single account, not counting cash balances.*/

   TreeMap<String,Double> getCashBalances()
   /* Returns an ordered Map of all account ID's and their cash balances. */

   TreeMap<String,Integer> getStockOwners( String ticker )
   /* Returns an ordered Map of account IDs and number of shares for each
      account that owns the stock identified by the ticker parameter. */

   List<Portfolio> getPortfoliosByCashBalances()
   /* Returns a list of all Portfolios, sorted in descending order
      by their cash balances. */

   List<Investment> getStocksByTicker()
   /* Returns a list all investments, sorted in ascending order by
```

```
ticker symbols.*/
```

What to Turn In

All of your classes must belong to the **cop4814.asg3** package. Upload a single zip file containing **only your .java files**, with no folders or other files. The zip file must be named after the last name and first initial of one of your team members. Both your team member names must appear in comments at the top of the AccountManage class file.

Turn in a complete printout of your source code, stapled in the upper-left corner, printed on white paper on one side in high-contrast black letters. (If your editor converts color fonts to grayscale, you will have to find a way to convert the text to pure black and white when printing.) On the last page, print your program's output, using the test program that I will provide.