**COMP 132 - Homework # 4**

**ArrayList Collections**

**Note: Uses CoinCollection.java and Coin.java, in the CoinCollection folder**

1. Give a line of code using the new operator to perform the following operation:

Declare a variable named myList and set it to hold a reference to a new ArrayList of DVD object references.

1. Complete the implementation of the CoinCollection class according to the specifications contained in its javadoc comments.
2. Draw the object diagram that would result from the following sequence of operations performed using the CoinCollection class.

1. Create a CoinCollection object named myColl with the collection name being “Loose Change”.

2. Add a Coin object named coin1 representing a good condition, 1932, penny that is worth $2.50.

3. Add a Coin object named coin2 representing a poor condition, 1850, nickel that is worth $12.75.

4. Add a Coin object named coin3 representing a mint condition, 1790, quarter that is worth $135.00.

1. How would you remove the 1850 nickel from the MyCoins collection? How would this change your object diagram? (You don't have to redraw it, just describe the change.)
2. Perform the following tasks using the CoinCollection class that you began working with above:

a. Add a getCoin method to the CoinCollection class. This method must accept an integer parameter indicating the position of the Coin to get. If the position specified by the parameter is valid, this method returns the requested Coin. If the position is not valid (i.e. it would cause an IndexOutOfBoundsException, this method prints an error message and returns null. You must also include a javadoc comment above this method describing what it does. Use the comments above the methods in the VideoStore class as a model.

b. Create test methods for all of the methods in your CoinCollection class (use the appropriate Testing Rules discussed in class.

1. What output, if any, would be produced by the following code snippets? (You need to figure these out by hand - do not run them on a computer.)

*a.* for (int i=3; i>=7; i++) {

System.out.println(i);

}

*b.* int x = 1;

for (int k=24; k>0; k=k-x) {

System.out.println(k);

x++;

}

1. Make the following additions to the CoinCollection project that you have been working with for the last several assignments. For each method, be sure that you use the specified name and that your method signatures are consistent with what is described below.

a. Add a method named printAllCoins. This method does not return a value and does not accept any parameters. When invoked it prints all of the coins in the collection. Be sure to include a javadoc comment describing your method.

b. Create a test method for your printAllCoins method.

1. Make the following additions to the CoinCollection class. For each method, be sure that you use the specified name and that your method signatures are consistent with what is described below.

a. Add a method named getTotalValue. This method returns a double value indicating the total value of all of the coins in the collection. It does not accept any parameters. Be sure to include a javadoc comment describing your method.

b. Create a test method for your getTotalValue method.

c. Add a method named countCoinsOlderThan that returns the number of Coins that were minted during or before a specified year. This method takes one parameter indicating the year. If the CoinCollection does not contain any Coins minted during or before the specified year, then this method should return 0. Be sure to include a Javadoc comment for your method.

d. Develop test methods for your countCoinsOlderThan method. Your tests must achieve statement coverage.