ITCS 1213
Lab 8

Purpose: To practice working with one-dimensional arrays of primitives. To build a class definition writing the toString( ), compareTo( ) and equals( ) methods, writing and using a copy constructor and using aggregation.

Step 1. In this exercise you will be manipulating data in a one-dimensional array. All the work for this step will be done in public static void main(String[ ] args).

This is how to declare an array of primitives:

```
int [ ] myNums = new int[10]; //this creates an array of 10 ints and
                              //all the values are initialized to zero.
```

Create a class named ArrayWork. In this class create the public static void main(String[ ] args) method.

Create and initialize an array of ten ints. Change the values from zero to other values. (you can use random numbers using the Random class or type in values)

a) Print the contents of the array.

b) Print the contents of the array in reverse order.

c) Print every other element of the array.

d) Remove the 5th element of the array, moving all the other elements after the 5th element down one slot. Assign a value of 0 to the last slot in the array. Print the contents of the array once you have done this:

```
example:
original contents:   3  5  6  2  7  8  1   13 12  15
         remove 7:
new contents:        3  5  6  2  8  1  13  12  15  0
```

e) Remove the 3rd element of the array, shifting all the elements after the 3rd element down one slot. Fill in the last slot with a 0. Print the contents of the array once you have done this.

```
example:
original contents:  3  5  6  2  8  1  13  12  15  0
         remove 6:
new contents:       3  5  2  8  1  13 12  15   0  0
```

f) Create a new array, twice the size of the original array. Copy the contents of your original array into this new expanded array. Print the new array to verify.

Step 2: new project (aggregation)

Create an Invoice class. The fields for the Invoice class are:

```
    String invoiceID
    String description
    double amount
    boolean isPaid
    Customer cust
```

nb: this will not compile until you have written the code for the Customer class.

The Invoice class must have two constructor methods,a toString( )method, compareTo( ) method (based on the amount field) and equals( ) method(based on the invoiceID field), in addition to methods to change the amount due and isPaid.(set methods)
The toString( ) method for the invoice must call the toString( ) method from the Customer class. (see the Student aggregation example from the lecture notes).

You also need to write a Customer class definition.
A Customer has the following fields:

```
String lastName
String firstName
String id
```

The Customer class must have a default constructor, a constructor that accepts arguments and a copy constructor. The Customer class must also have a toString( ) method.

Create a Driver that has main( ). In main( )create three Invoice instances. In order to create an Invoice object, you must first create Customer objects. (see the example of Student aggregation from lecture). Create at least two Customer objects.

Print the two Invoice objects to the screen using the toString( ) method of the Invoice class. The driver will compare the two Invoice instances using the compareTo( ) method, to determine which has a higher amount due.

Then determine if the two Invoice objects are equal, based on the invoice id using the equals( ) method of the Invoice class.