These practice exercises are not required, but working on them is strongly encouraged. Possible solutions will be released on the due dates indicated below.

Practice 07a – Java Concepts Chapter 07: Arrays & Array Lists
Due: Wed, Nov 5th.

Exercise P7.8 - AlternatingSum

Read the exercise & use the following code in your solution:

Use the following class in your solution:

```
/**
   This class computes the alternating sum
   of a set of data values.
*/
public class DataSet
{
   /**
      Constructs an empty data set.
   */
   public DataSet()
   {
      final int DATA_LENGTH = 100;
      data = new double[DATA_LENGTH];
      dataSize = 0;
   }

   /**
      Adds a data value to the data set.
      @param x a data value
   */
   public void add(double x)
   {
      if (dataSize >= data.length)
      {
         // make a new array of twice the size
         double[] newData = new double[2 * data.length];
         // copy over all elements from data to newData
         System.arraycopy(data, 0, newData, 0, data.length);
         // abandon the old array and store in data
         // a reference to the new array
         data = newData;
   }
   data[dataSize] = x;
   dataSize++;
}

/**
   Gets the alternating sum of the added data.
   @return sum the sum of the alternating data or 0 if no data has been added
*/
public double alternatingSum()
{
   . . .
}
   private double[] data;
   private int dataSize;
}
```

Use the following class as your tester class:

```
/**
   This program calculates an alternating sum.
*/
public class AlternatingSumTester
{
   public static void main(String[] args)
   {
      DataSet data = new DataSet();

      data.add(1);
      data.add(4);
      data.add(9);
      data.add(16);
      data.add(9);
      data.add(7);
      data.add(4);
      data.add(9);
      data.add(11);

      double sum = data.alternatingSum();
      System.out.println("Alternating Sum = " + sum);
      System.out.println("Expected: -2");
   }
}
```

Practice 07b – Java Concepts Chapter 07: Arrays & Array Lists
Due: Friday, Nov 7th.

Exercise P7.2 - Purse

Read the exercise. Implement a class `Purse`. A purse contains a collection of coins. For simplicity, we will only store the coin names in an `ArrayList<String>`. (We will discuss a better representation in Chapter 8)

Supply a method: `void addCoin(String coinName)`

Add a method `toString` to the `Purse` class that prints the coins in the purse in the format:
`Purse[Quarter,Dime,Nickel,Dime]`

Use the following class in your solution:

```
import java.util.ArrayList;

/**
   A purse holds a collection of coins.
*/
public class Purse
{
   /**
      Constructs an empty purse.
   */
   public Purse()
   {
      coins = new ArrayList<String>();
   }


  /**
      Adds a coin to the purse.
      @param coinName the coin to add
   */
   public void addCoin(String coinName)
   {
      . . .
   }

   /**
      Returns a string describing the object.
      @return a string in the format "Purse[coinName1,coinName2,...]"
   */
   public String toString()
   {
      . . .
   }

   private ArrayList<String> coins;
}
```

Use the following class as your tester class:

```
/**
   This class tests the Purse class.
*/
public class PurseTester
{
   public static void main(String[] args)
   {
      Purse p = new Purse();
      p.addCoin("Quarter");
      p.addCoin("Dime");
      p.addCoin("Nickel");
      p.addCoin("Dime");

      System.out.println(p.toString());
      System.out.println("Expected: Purse[Quarter,Dime,Nickel,Dime]");
   }
}
```