**Directions**: Launch BlueJ and load the project Z:\csa\csa\_exercises. *Create the project if it does not already exist.* We will create one class containing several arrays and methods.

Do not forget to test!!!

***Java Directions****: If you are doing this as a Java exercise, you will need to create a class file for each class and test it in a main method. Sample Java main method follows:*

*Public static void main(String[] args) {*

*}*

**ArrayLists (*Important – read this*)**

Java arrays are not very flexible. If you want an array that holds 10 elements to hold 20, you need to create a new array with a size of 20. The ArrayList class allows programmers to work with arrays that easily grow and shrink in size with simple method calls.

You must import a Java package in order to work with the ArrayList class. Study up on ArrayLists before you go further in this exercise. The examples you find, will have the import statement defined. ***Note****: ArrayLists cannot work with primitive types like int or double. They are typically used for Strings.*

**ArrayTest Class**

In this exercise, you will create the ArrayTest class. You will define several arrays and manipulate them in the methods you define.

1. **Import the java package required to work with ArrayLists.**
2. **Define the Array Instance Variables** – They are as follows:
   1. **intArray** – an empty integer array that can hold no more than 25 items.
   2. **stringArray -** an empty String array that can hold no more than 25 items.
   3. **stringArrayList** - an empty String ArrayList.

***Continued on next page***

1. **Create the constructor method** – The constructor method will populate the arrays with default values. Use for loops to populate the following arrays as defined below:
   1. **intArray –** populate this array with the numbers 1-25.
   2. **stringArray** – populate this array with 25 strings that say “stringArray Item: 1”, “stringArray Item: 2” etc.
   3. **stringArrayList** – populate this array with 25 strings that say “stringArrayList Item: 1”, “stringArrayList Item: 2” etc.

**Note:** All loops in this exercise must *dynamically* work with arrays of different sizes (different numbers of elements). *You should have already learned to retrieve the length of the array in a previous exercise. Look it up if you must.*

1. **Create the printArray()** **method** - This method will be used to print integer, String and ArrayList arrays, therefore it must be overloaded.*You should have already learned about method overloading in a previous exercise. Look it up if you must.*
   1. Create the version of the method that prints the values of a String array that is passed into it.
   2. Create the version of the method that prints the values of an integer array that is passed into it.
   3. Create the version of the method that prints the values of StringArray that is passed into it.
   4. Nothing is returned from any of the above methods.
   5. Test your method.
2. **Create the printArrays()** **method –** Create a method that calls the printArray() method one time for each array we are testing in this exercise.
   1. Check your output by calling printArrays() at the end of your constructor. Does it look good???

***Continued on next page***

1. **Create the updateArrays() method** – This method will modify our arrays. Code the following logic into this method.
   1. Modify intArray so that every odd number is multiplied by 10.
   2. Modify the third element in the stringArray to say “This is the third element in the string array”
   3. Modify the last element in the stringArray to say “This is the final element in the string array”.
      1. Your code must determine the array length in order to figure out what is the last element. **No magic numbers!**
   4. Modify the fifth element of the stringArrayList to say “*Yo, this is the fifth element dude”*.
   5. Modify the last element in the stringArrayList to say “Last element dude”.
      1. Your code must determine the array length in order to figure out what is the last element. **No magic numbers!**
   6. Insert an element at the fifth position in your stringArrayList that says “This has been inserted”
   7. Add an element to stringArrayList that says: “ArrayLists are really flexible. I like them”.
2. **Testing**
   1. Create an instance of the ArrayTest class using the name arrayObj for the reference variable.
      1. Two ways to test: Inside another method **or** within a main() method defined within this class.
   2. At the end of your constructor method, set the 26th element of intArray25 to the number 26 and test the class.
      1. Note the exception thrown by Java. *You need to know this for the exam.*
      2. Turn this line of code into a comment after you test it.
   3. Add the following lines of code after your instance has been created:
      1. arrayObj.printArrays();
      2. arrayObj.updateArrays();
      3. arrayObj.printArrays();
   4. Check the output to determine if everything worked.