Welcome to ArrayList and Objects Lab

Learning Outcomes

By the end of this lab:
- Be able to define chapter 9 terms
- Describe class Object and ArrayList.
- Draw a variable with a reference to an ArrayList holding objects.
- Trace, explain and write code using ArrayLists and Objects.
- Compare an array with ArrayList including length, size and capacity.
- Use enhanced for loop.

Exercise A: Terms

object (data and operations), reference, abstract data type (ADT), information hiding, encapsulation, ArrayList, element, ArrayList add/remove performance problem, enhanced for loop/for each loop

Exercise B: Trace and Explain

Example:
```java
ArrayList<String> fruit = new ArrayList<>();
fruit.add("apple");
fruit.add("orange");
fruit.add("grape");
```

Here is a diagram that shows the private internal detail of the ArrayList for this code example. Specifically, there is an array within the instance of ArrayList that holds the elements and is managed by the methods within the ArrayList class.

However, following the principle of encapsulation, since we are simply users of ArrayList, we can just show the publicly visible view that is relevant to understanding the examples below.
Draw a diagram showing memory for each to help explain what is happening.

1. `ArrayList<Double> nums;`

2. `ArrayList<Double> nums = new ArrayList<>();
   nums.add( 8.0); nums.add( 9.5); nums.add(10.1);`

3. `ArrayList<Character> chars = new ArrayList<>();
   chars.add('A');
   chars.add('B');
   chars.set(1, 'C');
   chars.add(0,'D');
   chars.remove(1);
   for ( int i = 0; i < chars.size(); i++) {
       System.out.print( chars.get(i) +", ");
   }

4. `ArrayList<Character> chars = new ArrayList<>();
   chars.add('B');
   chars.add('A');
   chars.add(1,'D');
   chars.add(0,'C');
   chars.set(1,'E');
   chars.remove(2);
   for ( char ch : chars) {
       System.out.print( ch +", ");
   }

5. `ArrayList<int[]> nums = new ArrayList<>();
   nums.add(new int[]{1,2,3});
   nums.add(new int[]{4,5,6});
   nums.add(new int[]{7,8,9});`

6. `ArrayList<ArrayList<Integer>> nums = new ArrayList<>();
   nums.add( new ArrayList<Integer>());
   nums.add( new ArrayList<Integer>());
   nums.add( new ArrayList<Integer>());
   nums.get(1).add(4);
   nums.get(2).add(5);`
nums.get(1).add(0,3);
nums.add( nums.get(1));
nums.get(3).add(1,6);

7. ArrayList<ArrayList<String>> nums = new ArrayList<>();
    nums.add( new ArrayList<String>());
    nums.add( new ArrayList<String>());
    nums.add( new ArrayList<String>());
    nums.get(0).add("The");
    nums.get(1).add("best");
    nums.get(1).add(0,"thing");
    nums.add( nums.get(0));
    nums.get(2).add("about");
    nums.add(1, nums.get(0));
    nums.get(1).add(1,"a");
    nums.get(2).add(1,"boolean");

8. ArrayList<Integer> list = new ArrayList<>();
    for (int i = 1; i <= 10; i++) {
        list.add(i);
    }
    for (int i = 0; i < list.size(); i++) {
        list.remove(i);
    }
    System.out.println(list);

9. ArrayList<Integer> list = new ArrayList<>();
    for (int i = 0; i < 4; i++) {
        System.out.println( list.get(i));
    }
    System.out.println( list.toString());

10. public static void methodA(ArrayList<Boolean> items) {
    items.add(true);
}
    public static void main(String []args) {
        ArrayList<Boolean> list = new ArrayList<>();
        methodA( list);
        System.out.println( list);
    }

11. public static void methodB(ArrayList<Boolean> items) {
    items = new ArrayList<Boolean>();
    items.add( true);
    return;
}
    public static void main(String []args) {
        ArrayList<Boolean> list = new ArrayList<>();
        methodB( list);
        System.out.println( list);
    }
methodB( list);
System.out.println( list);
}

Exercise C: Comparison
Describe the similarities and differences between the following array and ArrayList.

char[] list1 = new char[5];
list1[0] = 'a';
list1[1] = 'b';
System.out.println( "length: " + list1.length);

ArrayList<Character> list2 = new ArrayList<>();
list2.add( 'a');
list2.add( 'b');
list2.add( 'c');
System.out.println( "size: " + list2.size());

- How is capacity related to length and size?
- Is there any autoboxing occurring?
- What are ways one can keep track of which elements in an array are used and which are unused?
- Is there a method to determine the current capacity of the ArrayList?

Exercise D: Hashtag Count Chart
Ask the user for a hashtag and the count of that hashtag per day. Then show some summary statistics and a bar chart. Some starting code: Counter.java An example is:

Enter a hashtag: happy
Enter count of #happy tweets per day (negative to end).
200 213 43 25 19 100 -3
For 6 days minimum=19, maximum=213, average=100.000000
******************************************************************************(200)
******************************************************************************(213)
************(43)
***** (25)
**** (19)
************(100)
A * represents 4 tweets.

Exercise E: Count per Hashtag Chart
Ask the user for hashtags. Count the number of each using parallel ArrayLists. Then show some summary statistics and a bar chart. Some starting code: CountEachHashtag.java Example output:

Enter a list of hashtags ('end' to end):
love instagood photooftheday fashion beautiful happy love happy photooftheday love happy happy fashion
Exercise F: Debugging with Eclipse

Eclipse has a powerful debugger. Work through the Debugging Tutorial here:
Also available via the CS 200 Resources webpage.

Additional Learning Materials

When you have mastered everything in this lab, then you are welcome to learn from additional learning resources available on the web and beyond this course:
https://cs200-www.cs.wisc.edu/wp/learn-to-program-resources/

Note: Due to programs and zyBooks being individual work, it is Not appropriate to work on them during the Team Lab.

Lab designed by Jim Williams, Marc Renault. Yien Xu contributed an exercise.