

**CSCI 1301 – Programming Principles I**  
**Georgia Southern University**  
**Department of Computer Science**  
**Fall 2024**

**Lab 10**

**Point Value: 20 points**

**Due: Friday November 1, 2024, end of lab**

**Objectives**

1. Work with defining, creating, and using arrays.
2. Incorporate arrays into working programs that include methods, looping, conditional statements, and variables.
3. Use the debugger to effectively diagnose and fix errors in code.
4. Archive Java source code files and other documents into a zip file.

**Description**

Complete the following steps as described and upload requested materials to Folio in the appropriate dropbox in a single zip file. Submit Java source code files to Gradescope.

- Students should work in groups of two for this lab. When submitting your zip file, fill out the comment header and indicate both group member names.
- Your submitted zip file should be named Lab#LastName1LastName2.zip. For instance, Lab10JonesSmith. Any requested screenshots and/or answers to questions should be included in a separate Word document.
- All programs should have a comment header. That comment header should include information similar to:

```
/**
 * File: PAssign0.java
 * Class: CSCI 1301
 * Author: Christopher Williams
 * Created on: Jun 6, 2016
 * Last Modified: Aug 16, 2018
 * Description: Display three messages to the console
 */
```
- Before writing any code, for each problem, discuss with your partner how you plan to approach solving the given problem. Once you both agree on a way to attempt the problem, only then should code be written.
- Pay attention to names and any other requested material. Failure to include material will result in the loss of points as described below.

## Lab Problems

For all problems, use inline comments to describe major actions. Ask for assistance if needed.

1. Creating Arrays/Passing Arrays to Methods – Write a program named Lab10Prob01.java. In this program, create an array of 21 doubles with multiples of 7 using a for loop.

Create a method named `printArray()` that does not return anything and takes an array of doubles as a parameter. This method should print all values in the array to the Console one line at a time. You will use this method with your other problems to verify array contents.

In your main method, pass your created array to the `printArray()` method and verify that the elements in your array are printed to the Console in the order they are stored.

### Expected Output (some values excluded):

```
0.0
7.0
14.0
...
133.0
140.0
```

After passing the Gradescope tests, show a working version of this program to the instructor or TA before moving on.

**After instructor check-off, BOTH students should submit a final, identical version to Gradescope. This version is used for your final grade.**

2. Copying Arrays/Returning Arrays from Methods – Copy and paste your code into a new program named Lab10Prob02.java **but delete the code in your main method (leave the `printArray()` method)**. Write a method named `halfCopyArray()` that takes an array of doubles and returns an array that is “half” of the passed array’s elements as a new array. Assume that the “mid-point” or “half-way point” is the index that is the ceiling of the array’s length divided by 2 as shown below. Copy from the first element up to, and **excluding**, the value at the mid-point into the new array.

$$array_{mid} = \left\lceil \frac{array_{length}}{2} \right\rceil$$

For instance, given the array [18.7, -22.2, 43.4, 74.1, -25.5, 46.6, 27.8], the mid-point would be the ceiling of  $7 \text{ (length)}/2 = 3.5 \Rightarrow 4$ . This would result in elements 0-3 (18.7, -22.2, 43.4, 74.1) being copied into the new array.

Create the provided example array using the shortcut initializer. Create a “half-copy” of that array using the `halfCopyArray()` method and display it using `printArray()`.

**Expected Output:**

18.7  
-22.2  
43.4  
74.1

After passing the Gradescope tests, show a working version of this program to the instructor or TA before moving on.

**After instructor check-off, BOTH students should submit a final, identical version to Gradescope. This version is used for your final grade.**

3. Parallel Arrays – Recall that if two (or more) arrays are arranged so that related material is stored at the same index, that related material can be accessed within a for loop using the current value of the index.

```
String[] name = { "John Smith", "Mary Lewis", "Tom Jones", "Jane Brown" };  
double[] finalGrade = { 72.5, 97.5, 82.6, 85.7 };  
  
for (int i = 0; i < name.length; i++) {  
    System.out.printf("Name: %12s\tGrade: %.2f\n", name[i], finalGrade[i]);  
}
```

Write a Java program named `Lab10Prob03.java` that calculates the temperature difference for a given set of high and low temperatures and stores those differences in a third array. Print the “day” number (the index for that array) and temperature difference for days that had more than a 25-degree difference between high and low temps for that day.

Use the following values, in the following order, for the high and low temperatures (actual recorded values from Savannah/Hilton Head International Airport for October 17-27, 2024):

Highs: 66, 73, 77, 79, 81, 82, 83, 85, 85, 87, 82  
Lows: 41, 44, 49, 51, 49, 62, 59, 59, 57, 61, 63

These arrays are related by position. Day 0 (Oct 17) had a high of 66 and a low of 41, Day 1 (Oct 18) had a high of 73 and a low of 44, etc. For example, based on differences, Day 0 should not be printed as the high and low were not more than 25 degrees apart (they are exactly 25 degrees apart). You do not need to print the actual date; only which “day” number was valid.

Additionally, calculate and display the following across all temperatures:

- min low temperature

- average low temperature
- max high temperature
- average high temperature
- average difference

Finally, display a count of how many days there were during this span with a difference higher than the average.

#### **Expected Output:**

Days greater than 25 degree difference:

Day 1: 29

Day 2: 28

Day 3: 28

Day 4: 32

Day 7: 26

Day 8: 28

Day 9: 26

Statistics:

Min Low: 41

Average Low: 54.09

Max High: 87

Average High: 80.00

Average Difference: 25.91

Number of days with above-average difference: 7 out of 11

After passing the Gradescope tests, show a working version of this program to the instructor or TA before moving on.

**After instructor or TA check-off, BOTH students should submit a final, identical version to Gradescope. This version is used for your final grade.**

4. Demonstrate all working programs to the instructor or TA to receive credit for each problem. Turn in each problem inside the zip file detailed below.
5. Name the zip file Lab10LastName1LastName2.zip (e.g. Lab10JonesSmith) and submit that zip file to Folio. Both students must submit the exact same zip file for either to receive a grade.

#### **Grade Breakdown**

Demonstrate running application to instructor or TA:

Problem 1	4 points
Problem 2	6 points
Problem 3	8 points
Zipped source files (in Folio):	2 points
<hr/>	
Total Possible:	20 points

Last modified: November 4, 2024