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

Lab 9
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.
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, Lab9JonesSmith. 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.

!!--IMPORTANT--!!: For Problem 1, you **MUST** follow the provided instructions for Gradescope to correctly grade your submission.

1. Declaring and Creating Arrays – Write a program named Lab09Prob01.java. In this program you will create an array that is populated with 12 random whole numbers between -256 and 256, including both numbers as possible random numbers (inclusive on both sides).

   Recall the generic formula for specifying a range of values between a-b: a + Math.random() * (b-a). Make sure to correctly account for the inclusive part on the b end by adding 1 and add any appropriate casting. Do **not** use any Googled solution that uses the Java's built-in Random class.

   After creating the array, print the length of the loop using the array's built-in properties. Then, using a for loop, print each array element's value to the Console. Additionally, print the minimum value found in the array and its location.

   This program should work for any array of ints and all loop control and output values should be set appropriately (i.e. NOT hard-coded).

   !!--IMPORTANT--!!: Before submitting to Gradescope, add the following import statement:

   ```
   import gsu.Math;
   ```

   Expected Output:

   ```
   Length: 12

   118
   -45
   -149
   -85
   240
   -252
   238
   226
   229
   224
   -52
   -77

   Minimum value: -252 at index 5
   ```

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. Array Shortcut Initializer – Write a program named Lab08Prob02.java. In this program, create an array of 10 doubles using the array shortcut initializer. Populate this array with the following values (downloading the PDF might make copying easier): -12.6018020118 15.3437682821 145.8110691671 11.1681119916 82.0442710394 155.0005475009 -6.1179400421 120.1984991874 38.6575114628 77.1494972203.

   Using a for loop, print each array element and its index to the Console with any decimal values being restricted to 5 decimal places. Additionally, print the maximum value found to in the array and its location to 5 decimal places.

   This program should work for any array of doubles and all loop control and output values should be set appropriately (i.e. NOT hard-coded).

   Expected Output:

   ```
   Index 0 of 9: -12.60180
   Index 1 of 9: 15.34377
   Index 2 of 9: 145.81107
   Index 3 of 9: 11.16811
   Index 4 of 9: 82.04427
   Index 5 of 9: 155.00055
   Index 6 of 9: -6.11794
   Index 7 of 9: 120.19850
   Index 8 of 9: 38.65751
   Index 9 of 9: 77.14950

   Maximum Value: 155.00055 at index 5
   ```

   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. Processing Arrays – Copy and paste your array from Problem 2 into a new program named Lab08Prob03.java. Write code to process the array to determine and display the minimum value, the maximum value, and the average value (all restricted to 3 decimal places) to the Console.

Make intelligent choices about the initial values for your minimum and maximum and initial value for your loop control variable.

After printing those calculated values, then print a count of how many values in the array were above the average and a count of values below the average.

Expected Output:

```
Maximum value: 155.001
Minimum value: -12.602
Average value: 62.665
Values above average: 5
Values below average: 5
```

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 Lab9LastName1LastName2.zip (e.g. Lab9JonesSmith) 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	6 points
Problem 2	6 points
Problem 3	6 points
Zipped source files (in Folio):	2 points
Total Possible:	20 points

Last modified: October 28, 2024