

# Lab #7

## CSE110 - Arizona State University

### Topics

- Basic arrays

### Coding Guidelines

- Give identifiers semantic meaning and make them easy to read (examples numStudents, grossPay, etc).
- Keep identifiers to a reasonably short length.
- Use upper case for constants. Use title case (first letter is upper case) for classes. Use lower case with uppercase word separators for all other identifiers (variables, methods, objects).
- Use tabs or spaces to indent code within blocks (code surrounded by braces). This includes classes, methods, and code associated with ifs, switches and loops. Be consistent with the number of spaces or tabs that you use to indent.
- Use white space to make your program more readable.
- Use comments after the ending brace of classes, methods, and blocks to identify to which block it belongs.

### Assignment/Lab Documentation

At the beginning of each programming assignment you must have a comment block with the following information:

```
/*
// AUTHOR: your name
// FILENAME: title of the source file
// SPECIFICATION: description of the program
// FOR: CSE 110- Lab #7
// TIME SPENT: how long it took you to complete the assignment
//-----*/
```

### Getting Started

Create a class called **Lab7**. Use the same setup for setting up your class and main method as you did for the previous assignments. Be sure to name your file **Lab7.java**.

### Hints

- (Page 250) An array will be used in this program.
- The lecture and code example videos are also good references for this lab.

## Task Overview

The overall goal of this lab is to allow the user to enter five integers into an array. The program will then print out each of these integers + 1.

### Part 1: Initial Setup

The following needs to be done for the initial setup:

- Create a `Scanner` object - be sure to import the `Scanner` class
- Declare a constant `int` called `SIZE` - set it equal to 5
- Create an array of `ints` called array of size `SIZE`

Note: See the table on page 252 if you need help creating your array.

### Part 2: Filling the Array

Using a loop, iterate through the array and use the `Scanner` class to read `ints` into all five positions in the array. A prompt for input is optional. See the lecture videos for examples of this being done. If this part is completed successfully, your array should contain five elements after the user enters five integers.

### Part 3: Displaying the Output

To separate the outputs, include a `println` statement stating something along the lines of:

Processing each array element...

Now, write another loop which prints out  $1 +$  each of the array's elements on a separate line. For example, if one of the elements is 4, your program should print a 5. See below for a full example.

#### Note

- It isn't necessary to prompt a user to input integers in the beginning. It isn't a problem if you do, though.
- It is possible to achieve the output of this lab without using arrays and with only one loop. However, that would defeat its purpose. Please use an array and please write out the two loops to complete this lab for full credit.
- Labs are not graded by a program, so you do not need to spend a large amount of time making the output match perfectly with the sample below. Do, however, make sure your output is reasonable. The goal here is for you to demonstrate that you understand the underlying concepts.

#### Sample Output

Below is an example of what your output should roughly look like when this lab is completed. All text in bold represents user input.

Sample Run 1:

**0**  
**1**  
**2**  
**3**  
**4**

Processing each array element...

1  
2  
3  
4  
5

## Sample Run 2:

-5  
6  
0  
1  
1  
Pr  
-4  
7  
1  
2  
2

Processing each array element...

-4  
7  
1  
2  
2

## *Submission*

Submit your `Lab7.java` to the Submission Server. Go to the Submission Server site located on the course website, login, then click on Lab Submissions in the left frame. Choose Lab7 from the dropdown box, click on the browse button and find where you saved your `Lab7.java` on your computer. Upload the file to the site and then click on the Submit button.

Your file will be submitted and a screen will show up displaying if your program compiled and what your output is when run on some sample input.

You should then check to make sure that the actual file submitted properly and is readable to the grader. To do so click on Grades in the frame on the left of the page and then click on the 0 underneath Lab7. You will again see that your program compiled and the sample output, but you should scroll down to the bottom of the screen and make sure your file is readable as well.

**Important Note:** You may resubmit as many times as you like until the deadline, but we will only mark your last submission.