

# CS102A Introduction to Computer Programming

## Fall 2020

### Lab 5



#### Credit

The source code and document description are designed by ZHU Yueming.

## Objectives

1. Learn how to declare, create, and initialize arrays.
2. Learn how to copy and print arrays in `for` loops.
3. Learn how to use arrays to implement simple algorithms.

## 1 Prework

Copy the following code in your program. It creates two arrays and uses two different ways to print their elements:

```
1 int array1[] = {1, 2, 3, 4, 5};
2 int array2[] = new int[5];
3 array2[0] = 6;
4 array2[1] = 7;
5 array2[2] = 8;
6 array2[3] = 9;
7 array2[4] = 10;
8
9 for(int i=0; i<array1.length; i++) {
```

```

10     System.out.print(array1[i] + "\t");
11 }
12 System.out.println();
13 for(int e:array2) {
14     System.out.print(e + "\t");
15 }
16 System.out.println();

```

Add the following code. It declares another array and assigns it to `null`. See what the difference is between the two arrays:

```

1 int array3[] = null;
2 System.out.println(array3);
3
4 array3 = array2;
5 System.out.println(array3);

```

Add the following code. Notice when the elements of `array3` are changed:

```

1 for(int e:array3) {
2     e = 1;
3 }
4 System.out.println("array3: " + Arrays.toString(array3));
5 for(int i=0; i<array3.length; i++) {
6     array3[i] = 1;
7 }
8 System.out.println("array3: " + Arrays.toString(array3));

```

Run the following code. Why are the elements of `array2` changed?

```

1 System.out.println("array2: " + Arrays.toString(array2));

```

Finally, run the following code:

```

1 char[] Array4 = {'a', 'b', 'c'};
2 System.out.println(Array4);

```

## 2 Exercises

### 2.1 Exercise 1

In this exercise, you will practice performing the following basic array operations:

1. Declare and create an array named `myList1` with `n` ( $0 < n < 20$ ) elements of type `double`.
2. Initialize `myList1` with input values and make a copy of `myList1` named `myList2`.
3. Shift the elements in `myList1` to the left by one position.
4. Print the elements in `myList1` and `myList2`.

Sample input and output:

```
> java Lab5E1
Enter the length of myList1: 8
Enter 8 values: 2.5 5.5 3.4 6.4 7.7 2.2 8.9 0.2
myList1: 5.5 3.4 6.4 7.7 2.2 8.9 0.2 2.5
myList2: 2.5 5.5 3.4 6.4 7.7 2.2 8.9 0.2
```

### 2.2 Exercise 2

Write a program that computes the average score of 10 students. All scores must lie within the range of  $[0, 100]$ . The average score should be computed after removing the lowest and highest scores.

Sample input and output:

```
Please input the scores of these 10 students: 88.3 99 45 78 67.5
98.4 23.5 65.5 82 85.4
The average score is 76.26
```

### 2.3 Exercise 3

Write a program that compares two arrays of the same size. Let the user input the array size and every element of the two arrays. Two arrays are considered equal if and only if:

1. Both arrays contain the same number of elements; and
2. All corresponding pairs of elements in the two arrays are equal.

Sample input and output:

```
> java Lab5P3
Enter the length of array:4
Enter the 1st integer array of size 4:1 2 3 4
Enter the 2nd integer array of size 4:1 2 3 4
The two arrays are equal.

> java Lab5P3
Enter the length of array:3
Enter the 1st integer array of size 4:1 2 3
Enter the 2nd integer array of size 4:3 2 1
The two arrays are not equal.
```

## 2.4 Exercise 4

Write a program that reads a sequence of integers with values between 1 and 100. A zero indicates the end of the sequence. The program will then print the occurrences of each.

Sample input and output:

```
Enter the integers between 1 and 100: 22 33 35 34 99 87 45 34 23
78 45 33 11 23 87 34 76 0
11 occurs 1 time
22 occurs 1 time
23 occurs 2 times
33 occurs 2 times
34 occurs 3 times
35 occurs 1 time
45 occurs 2 times
76 occurs 1 time
78 occurs 1 time
```

```
87 occurs 2 times
99 occurs 1 time
```

## 2.5 Exercise 5

Write a program that sorts a sequence of integers in ascending order. The user will first input the size of the sequence, then the integers. The program will finally print the sorted sequence.

Sample input and output:

```
How many numbers you will input: 10
3 5 2 99 44 54 23 46 87 56
2 3 5 23 44 46 54 56 87 99
```

## 2.6 Exercise 6

Write a program that prompts the user to input  $n$  integers from 1 to 1000 in ascending order. Let  $\mu$  be the average value of all the integers. Count how many pairs of integers have an average value greater than  $\mu$ .

You are asked to design your program to accomplish this task as fast as possible. To estimate the running time of your algorithm, you may calculate the difference `current2-current1` as follows:

```
1 long current1=System.currentTimeMillis();
2 /* your algorithm */
3 long current2=System.currentTimeMillis();
4 System.out.printf("Your program ran for %.3f seconds", (current2-
   current1)/1000.0d);
```

Sample input and output:

```
Enter how many numbers: 5
Enter 5 numbers:
1 2 3 4 5
average=3.0
The number of integer pairs is 4
```

Your program ran for 0.004 seconds

Enter how many numbers: 30

Enter 30 numbers:

2 3 5 6 9 10 12 13 15 16 23 55 66 77 89

101 220 221 222 255 277 280 290 300 303

400 420 455 500 520

average=172.16666666666666

The number of integer pairs is 194

Your program ran for 0.004 seconds