

# Intro to Java Week 3 Coding Assignment

Points possible: 70

Category	Criteria	% of Grade
Functionality	Does the code work?	25
Organization	Is the code clean and organized? Proper use of white space, syntax, and consistency are utilized. Names and comments are concise and clear.	25
Creativity	Student solved the problems presented in the assignment using creativity and out of the box thinking.	25
Completeness	All requirements of the assignment are complete.	25

**Instructions:** In Eclipse, or an IDE of your choice, write the code that accomplishes the objectives listed below. Ensure that the code compiles and runs as directed. Take screenshots of the code and of the running program (make sure to get screenshots of all required functionality) and paste them in this document where instructed below. Create a new repository on GitHub for this week's assignments and push this document, with your Java project code, to the repository. Add the URL for this week's repository to this document where instructed and submit this document to your instructor when complete.

## Coding Steps:

1. Create an array of int called ages that contains the following values: 3, 9, 23, 64, 2, 8, 28, 93.
  - a. Programmatically subtract the value of the first element in the array from the value in the last element of the array (i.e. do not use ages[7] in your code). Print the result to the console.
  - b. Add a new age to your array and repeat the step above to ensure it is dynamic (works for arrays of different lengths).
  - c. Use a loop to iterate through the array and calculate the average age. Print the result to the console.
2. Create an array of String called names that contains the following values: "Sam", "Tommy", "Tim", "Sally", "Buck", "Bob".
  - a. Use a loop to iterate through the array and calculate the average number of letters per name. Print the result to the console.

- b. Use a loop to iterate through the array again and concatenate all the names together, separated by spaces, and print the result to the console.
3. How do you access the last element of any array?
4. How do you access the first element of any array?
5. Create a new array of int called nameLengths. Write a loop to iterate over the previously created names array and add the length of each name to the nameLengths array.
6. Write a loop to iterate over the nameLengths array and calculate the sum of all the elements in the array. Print the result to the console.
7. Write a method that takes a String, word, and an int, n, as arguments and returns the word concatenated to itself n number of times. (i.e. if I pass in "Hello" and 3, I would expect the method to return "HelloHelloHello").
8. Write a method that takes two Strings, firstName and lastName, and returns a full name (the full name should be the first and the last name as a String separated by a space).
9. Write a method that takes an array of int and returns true if the sum of all the ints in the array is greater than 100.
10. Write a method that takes an array of double and returns the average of all the elements in the array.
11. Write a method that takes two arrays of double and returns true if the average of the elements in the first array is greater than the average of the elements in the second array.
12. Write a method called willBuyDrink that takes a boolean isHotOutside, and a double moneyInPocket, and returns true if it is hot outside and if moneyInPocket is greater than 10.50.
13. Create a method of your own that solves a problem. In comments, write what the method does and why you created it.

### Screenshots of Code:

```
package week3;

public class Week3JavaAssignment {

    public static void main(String[] args) {

        // Problems 1a and 1b

        int[] ages = {3, 9, 23, 64, 2, 8, 28, 93, 10};
        System.out.println(ages[ages.length - 1] - ages[0]);

        // Problem 1c

        int sum = 0;
        double average = 0;
        for (int i = 0; i < ages.length; i++) {
            sum += ages[i];
        }
        average = (double)sum / ages.length;
        System.out.println("Average age = " + average);
    }
}
```

// Problem 2a

```
String[] names = {"Sam", "Tommy", "Tim", "Sally", "Buck", "Bob"};
int charSum = 0;
double avgChars = 0;
for (int i = 0; i < names.length; i++) {
    charSum += names[i].length();
}
avgChars = (double)charSum / names.length;
System.out.println("Average number of letters = " + avgChars);
```

// Problem 2b

```
for (int i = 0; i < names.length; i++) {
    System.out.print(names[i] + " ");
} System.out.println();
```

// Problem 3 and 4

// The last element in an array is accessed by names[names.length - 1]  
// The first element in an array is accessed by names[0]

// Problem 5

```
int[] nameLengths = new int[names.length];
for (int i = 0; i < names.length; i++) {
    nameLengths[i] = names[i].length();
}
```

// Problem 6

```
int nameSum = 0;
for (int i = 0; i < names.length; i++) {
    nameSum += nameLengths[i];
}
System.out.println("Sum of all letters = " + nameSum);
```

// Problem 7

```
String word = "Hello";
int n = 3;
concatenateWord(word, n);
```

// Problem 8

```
String firstName = "Charlie";
String lastName = "Dukes";
createFullName(firstName, lastName);
```

// Problem 9

```
int[] numbers = {23, 42, 43};
isGreaterThan100(numbers);
```

```
// Problem 10
```

```
double[] elements = {22.2, 33.3, 44.4};  
returnAverage(elements);
```

```
// Problem 11
```

```
double[] numbers1 = {12.3, 45.6, 78.9};  
double[] numbers2 = {22.3, 55.6, 88.9};  
isArrayAverageGreater(numbers1, numbers2);
```

```
// Problem 12
```

```
boolean isHotOutside = true;  
double moneyInPocket = 15.00;  
willBuyDrink(isHotOutside, moneyInPocket);
```

```
// Problem 13
```

```
// Given the amount of hours in a day that the user has been playing New World,  
// return some advice for the gamer i.e., 0-3 hours returns "Keep up the grind!",  
// 4-7 hours returns "Go grab some food.", 8+ hours returns "Go touch some grass..."  
// This will keep us fellow gamers held accountable :)
```

```
double hoursPlayed = 8.5;  
whatShouldIdo(hoursPlayed);
```

```
// Method for Problem 7
```

```
private static String concatenateWord(String word, int n) {  
    String newWord = "";  
    for (int i = 0; i < n; i++) {  
        newWord += word;  
    }  
    return newWord;  
}
```

```
// Method for Problem 8
```

```
private static String createFullName(String firstName, String lastName) {  
    return (firstName + " " + lastName);  
}
```

```
// Method for Problem 9
```

```
private static boolean isGreaterThan100(int[] numbers) {  
    int sum = 0;  
    for (int i = 0; i < numbers.length; i++) {  
        sum += numbers[i];  
    }  
    if (sum > 100) {  
        return true;  
    } else {  
        return false;  
    }  
}
```

```

// Method for Problem 10
private static double returnAverage(double[] elements) {
    double sum = 0;
    for (int i = 0; i < elements.length; i++) {
        sum += elements[i];
    }
    return (sum / elements.length);
}

// Method for Problem 11
private static boolean isArrayAverageGreater(double[] numbers1, double[] numbers2) {
    if (returnAverage(numbers1) > returnAverage(numbers2)) {
        return true;
    } else {
        return false;
    }
}

// Method for Problem 12
private static boolean willBuyDrink(boolean isHotOutside, double moneyInPocket) {
    if (isHotOutside == true && moneyInPocket > 10.50) {
        return true;
    } else {
        return false;
    }
}

// Method for Problem 13
private static void whatShouldIdo(double hoursPlayed) {
    if (hoursPlayed <= 3) {
        System.out.println("Keep up the grind!");
    } else if (hoursPlayed <= 7) {
        System.out.println("Go grab some food.");
    } else {
        System.out.println("Go touch some grass...");
    }
}
}

```

---

### Screenshots of Running Application:

```

7
Average age = 26.666666666666668
Average number of letters = 3.8333333333333335
Sam Tommy Tim Sally Buck Bob
Sum of all letters = 23
Go touch some grass...

```

### URL to GitHub Repository:

<https://github.com/DukesGuy/JavaProjects/tree/master/week3>

