

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:

```
*week3CodingAssignment.java X
1 import java.util.Arrays;
2
3 public class week3CodingAssignment {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7
8         // 1. Create an array of int called ages
9         //that contains the following values: 3, 9, 23, 64, 2, 8, 28, 93.
10
11         int[] ages = {3, 9, 23, 64, 2, 8, 28, 93};
12
13         //a. Programmatically subtract the value of the first element
14         //in the array from the value in the last element of the array
15         //(i.e. do not use ages[7] in your code). Print the result to the console.
16
17         int subtraction = (ages[ages.length - ages.length] - ages[ages.length-1]);
18
19         System.out.println("1.a " + subtraction);
20
21         //b. Add a new age to your array and repeat the step above to ensure
22         //it is dynamic (works for arrays of different lengths).
23
24         int[] newAge = new int[ages.length];
25
26         System.arraycopy(ages, 0, newAge, 0, ages.length);
27
28         newAge[newAge.length - 1] = 2;
29
30         int subtraction2 = (newAge[newAge.length - newAge.length] - newAge[newAge.length-1]);
31
32         System.out.println("1.b " + subtraction2);
33
34         //c. Use a loop to iterate through the array and calculate the average
35         //age. Print the result to the console.
36
37         int add = 0;
38
39         for (int i = 0; i < newAge.length; i++) {
40             add += newAge[i];
41         }
42         double avg = add / newAge.length;
43
44         System.out.println("1.c " + avg);
45
46         //2. Create an array of String called names that contains the following
47         // values: "Sam", "Tommy", "Tim", "Sally", "Buck", "Bob".
48
49         String[] names = {"Sam", "Tommy", "Tim", "Sally", "Buck", "Bob"};
50
51         // a. Use a loop to iterate through the array and calculate the average number
52         // of letters per name. Print the result to the console.
53         double sum = 0;
54         for (int i = 0; i < names.length; i++) {
55             sum += names[i].length();
56         }
57         avg = sum / names.length;
58         System.out.println("2.a " + avg);
59
60     }
```

```

60 // b. Use a loop to iterate through the array again and concatenate all the
61 // names together, separated by spaces, and print the result to the console.
62
63 String namesTogether = "";
64 for (int i = 0; i < names.length; i++) {
65     namesTogether += names[i] + " ";
66     System.out.println("2.b) " + namesTogether);
67 }
68
69
70 //3. How do you access the last element of any array?
71
72 System.out.println("3. To access the last element of an array use array[array.length - 1]");
73
74 //4. How do you access the first element of any array?
75
76 System.out.println("4. To access the first element of an array use array[0]");
77
78 //5. Create a new array of int called nameLengths. Write a loop to iterate
79 // over the previously created names array and add the length of each name to
80 // the nameLengths array.
81
82 int[] nameLengths = new int [names.length];
83 for (int i = 0; i < names.length; i++) {
84     nameLengths[i] = names[i].length();
85 }
86
87 //6. Write a loop to iterate over the nameLengths array and calculate the sum
88 // of all the elements in the array. Print the result to the console.
89 int sumOfElements = 0;
90 for (int i = 0; i < nameLengths.length; i++) {
91     sumOfElements += nameLengths[i];
92 }
93
94 System.out.println("6. " + sumOfElements);
95
96 //7. Write a method that takes a String, word, and an int, n, as arguments and
97 // returns the word concatenated to itself n number of times. (i.e. if I pass in
98 // "Hello" and 3, I would expect the method to return "HelloHelloHello").
99
100 System.out.println("7. " + multiples("Hello", 3));
101
102 //8. Write a method that takes two Strings, firstName and lastName, and returns
103 // a full name (the full name should be the first and the last name as a String
104 // separated by a space).
105
106 System.out.println("8. " + fullName("Bat", "Man"));
107
108 //9. Write a method that takes an array of int and returns true if the sum of all
109 // the ints in the array is greater than 100.
110
111 int[] test= {1,2,3};
112
113 System.out.println("9. " + greaterThan100(test));
114
115 //10. Write a method that takes an array of double and returns the average of all
116 // the elements in the array.
117

```

```

118     double[] arrayD = {0.1, 0.2, 0.3, 0.4};
119     System.out.println("10. " + dAvg(arrayD));
120
121     //11. Write a method that takes two arrays of double and returns true if the
122     // average of the elements in the first array is greater than the average of the
123     // elements in the second array.
124
125     double[] arrayD2 = {0.5, 0.6, 0.7, 0.8};
126     System.out.println("11. " + firstVsLast(arrayD, arrayD2));
127
128     //12. Write a method called willBuyDrink that takes a boolean isHotOutside, and
129     // a double moneyInPocket, and returns true if it is hot outside and if
130     // moneyInPocket is greater than 10.50.
131
132     boolean isHotOutside = true;
133     double moneyInPocket = 10.6;
134     System.out.println("12. " + willBuyDrinks(isHotOutside, moneyInPocket));
135
136     //13. Create a method of your own that solves a problem. In comments, write
137     // what the method does and why you created it.
138
139     int che = 100;
140
141     System.out.println("13. Your new checking acoount balance is " + bank(che));
142
143     //13 Ans: This method is used to determine the difference between a persons bills
144     // and and how much money they are getting paid. If the person has more bills than
145     //they there check can cover then the person will need to make up the extra from
146     //there checking account. If the check is higher than the bills than the person can add the
147     // difference to there checking account.
148 }
149
150
151 public static String multiples(String word, int n) {
152     String multiples = "";
153     for (int i = 0; i < n; i++) {
154         multiples += word;
155     }
156
157     return multiples;
158 }
159
160
161 public static String fullName(String first, String last) {
162     return first + " " + last;
163 }
164
165 public static boolean greaterThan100(int[] arra) {
166     int sum = 0;
167     for (int number : arra) {
168         sum += number;
169     }
170
171     return sum > 100;
172 }
173
174 public static double dAvg(double[] arr) {
175     double sum = 0;
176     for (double number : arr) {

```

```

177         sum += number;
178     }
179     return sum / arr.length;
180 }
181
182 public static boolean firstVsLast(double[] first, double[] second) {
183     double sum1 = 0;
184     double sum2 = 0;
185     for (double numb1 : first) {
186         sum1 += numb1;
187     }
188     for (double numb2 : second) {
189         sum2 += numb2;
190     }
191     return sum1 > sum2;
192 }
193
194 public static boolean willBuyDrinks(boolean isHotOutside, double moneyInPocket) {
195
196
197     return (isHotOutside == true && moneyInPocket > 10.50);
198 }
199
200 public static int bank(int checking) {
201     int check = 600;
202     int bills = 550;
203     int difference = check - bills;
204     if (difference > 0) {
205         return Math.abs(difference + checking);
206     }
207
208
209     return Math.abs(checking + difference);
210 }
211
212 }
213

```

Screenshots of Running Application:

```
Problems @ Javadoc Declaration Console X
<terminated> week3CodingAssignment [Java Application] C:\Program Files\Java\jdk-11.0.14\bin\javaw.exe (May 20, 2022, 4:21:16 PM – 4:21:17 PM) [pid: 17808]
1.a) -90
1.b) 1
1.c) 17.0
2.a) 3.8333333333333335
2.b) Sam |
2.b) Sam Tommy
2.b) Sam Tommy Tim
2.b) Sam Tommy Tim Sally
2.b) Sam Tommy Tim Sally Buck
2.b) Sam Tommy Tim Sally Buck Bob
3. To access the last element of an array use array[array.length - 1]
4. To access the first element of an array use array[0]
6. 23
7. HelloHelloHello
8. Bat Man
9. false
10. 0.25
11. false
12. true
13. Your new checking acoount balance is 150
```

URL to GitHub Repository:

<https://github.com/robertspwork/Week3CodingAssignment-ArraysAndMethods.git>