

# Intro to Java Week 3 Coding Assignment

**Points possible:** 70

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

```

App.java 33 | Console
1 package com.lisasmith.week3;
2
3 public class App {
4
5     public static void main(String[] args) {
6
7         /* Requirement 1
8          * Create an array of int called ages that contains the
9          * following values: 3,9,23,64,2,8,28,93.
10         */
11
12         System.out.println("-----");
13         System.out.println("Intro To Java Week 3 Coding Assignment");
14         System.out.println("-----");
15         System.out.println();
16         System.out.println("Requirement 1:");
17         System.out.println("-----");
18         int[] ages = {3,9,23,64,2,8,28,93};
19         System.out.print("Elements in the ages array are: ");
20         for (int i=0; i<ages.length; i++) {
21             System.out.print(ages[i] + " ");
22         }
23         System.out.println();
24         System.out.println();
25
26         /* Requirement 1(a)
27          * Programmatically subtract the value of the first element in
28          * the array from the value in the last element of the array
29          * (i.e. do not use "ages[7]" in your code).
30          * Print the result to the console.
31         */
32         int diffOfAges = ages[ages.length-1]- ages[0];
33         System.out.println("Requirement 1(a):");
34         System.out.println("-----");
35         System.out.println("The first array element is: " + ages[0]);
36         System.out.println("The last array element is: " + ages[ages.length-1]);
37         System.out.println();
38         System.out.println("The difference in the ages array between");
39         System.out.println("the last array element & the first array element is: "+ diffOfAges);
40         System.out.println();
41         System.out.println();
42
43         /* Requirement 1(b)
44          * Add a new age to your array and repeat the step above
45          * to ensure that it is dynamic (works for arrays of
46          * different lengths).
47         */
48         int[] ages2 = {3,9,23,64,2,8,28,93,100};
49         int diffOfAges2 = ages2[ages2.length-1]- ages2[0];
50
51         System.out.println("Requirement 1(b):");
52         System.out.println("-----");
53
54         System.out.print("Elements in the ages2 array are: ");
55         for (int i=0; i<ages2.length; i++) {
56             System.out.print(ages2[i] + " ");
57         }
58         System.out.println();
59         System.out.println();
60         System.out.println("The first array element is: " + ages2[0]);
61         System.out.println("The last array element is: " + ages2[ages2.length-1]);
62         System.out.println();
63         System.out.println("The difference in the ages2 array (e.g. the ages array plus one element)");
64         System.out.println("between the last array element & the first array element is: "+ diffOfAges2);
65         System.out.println();
66         System.out.println();
67
68         /*
69          * Requirement 1(c)
70          * Use a loop to iterate through the array and calculate the average age.
71          * Print the result to the console.
72         */
73
74         int sumAge = 0;
75         int averageAge = 0;
76         for (int num : ages ) {
77             sumAge += num;
78         }
79         averageAge = sumAge/ages.length;
80         System.out.println("Requirement 1(c):");
81         System.out.println("-----");
82         System.out.print("The average of all of the elements in the ages array is: ");
83         System.out.println(averageAge);
84         System.out.println();
85
86         /*

```

Outline com.lisasmith.week3

▼ C App

- S main(String[]) : void
- S multiplyConcat(String, int) : String
- S createFullName(String, String) : String
- S isGreaterThanOneHundred(int[]) : boolean
- S averageDouble(double[]) : double
- S isAverageGreaterThan(double[], double[]) : boolean
- S willBuyDrink(boolean, double) : boolean
- S returnCorrectChange(double, double) : double
- S addItemPrices(double[]) : double
- S pointOfSale(double, double) : boolean

```

87
88         /*
89          * Do the same calculation for the ages2 array
90         */
91         sumAge = 0;
92         int averageAge2 = 0;
93         for (int num : ages2 ) {
94             sumAge += num;
95         }
96         averageAge2 = sumAge/ages2.length;
97         System.out.print("The average of all of the elements in the ages2 array is: ");
98         System.out.println(averageAge2);
99         System.out.println();
100
101        /* Requirement 2.
102           * Create an array of String called names that contains the following values:
103           * "Sam", "Tommy", "Tim", "Sally", "Buck", "Bob".
104           */
105        String[] names = {"Sam", "Tommy", "Tim", "Sally", "Buck", "Bob"};
106        System.out.println("Requirement 2:");
107        System.out.println("-----");
108        System.out.print("The length of the names array is: ");
109        System.out.println(names.length);
110        System.out.print("The names in the names array are: ");
111        for (int i = 0; i < names.length; i++) {
112            System.out.print(names[i] + " ");
113        }
114        System.out.println();
115        System.out.println();
116
117        /* Requirement 2(a)
118           * Use a loop to iterate through the array and
119           * calculate the average number of letters per name
120           * Print the result to the console.
121         */
122
123        int sum = 0;
124        int avgLengthOfName = 0;
125        for (int i = 0; i < names.length; i++) {
126            sum += names[i].length();
127        }
128        avgLengthOfName = sum/names.length;
129
130        System.out.println("Requirement 2(a):");
131        System.out.println("-----");
132        System.out.print("The average number of letters per name in the names array is: ");
133        System.out.println(avgLengthOfName);
134        System.out.println();
135        System.out.println();
136
137        /* Requirement 2(b)
138           * Use a loop to iterate through the array again &
139           * concatenate all the names together, separated by spaces, &
140           * print the results to the console.
141         */
142
143        String concatenatedName = "";
144        for (String name : names) {
145            concatenatedName += (name + " ");
146        }
147        System.out.println("Requirement 2(b):");
148        System.out.println("-----");
149        System.out.print("The concatenation of the elements in the names array with spaces is: ");
150        System.out.println(concatenatedName);
151        System.out.println();
152        System.out.println();
153
154
155        /* Requirement 3
156           * How do you access the last element of any array?
157         */
158        int lastAgeOfAges = ages[ages.length-1];
159        String lastNameOfNames = names[names.length-1];
160        System.out.println("Requirement 3:");
161        System.out.println("-----");
162        System.out.println("Using our declared arrays, ages & names: and");
163        System.out.println("the index <arrayName>.arrayName.length-1");
164        System.out.println("-----");
165        System.out.print("The last element of the ages array is: " + lastAgeOfAges);
166        System.out.println();
167        System.out.print("The last element of the names array is: " + lastNameOfNames);
168        System.out.println();
169        System.out.println();

```

```

178 /* Requirement 4
179 * How do you access the first element of any array?
180 */
181 int firstAgeOfAges = ages[0];
182 String firstNameOfNames = names[0];
183 System.out.println("Requirement 4:");
184 System.out.println("-----");
185 System.out.println("Using our declared arrays, ages & names:");
186 System.out.println("   and the index <arrayName>[0]");
187 System.out.println("-----");
188 System.out.println("The first element of the ages array is: " + firstAgeOfAges);
189 System.out.println("The first element of the names array is: " + firstNameOfNames);
190 System.out.println();
191 System.out.println();
192
193 /*
194 * Create a new array of int called nameLengths.
195 * Write a loop to iterate over the previously created names array &
196 * add the length of each name to the nameLengths array.
197 */
198
199 int[] nameLengths = new int[names.length];
200 for (int i=0; i < names.length; i++) {
201     nameLengths[i] = names[i].length();
202 }
203 System.out.println();
204 System.out.println();
205 System.out.println();
206
207 /*
208 * Requirement 5
209 * Write a loop to iterate over the nameLengths array &
210 * calculate the sum of all the elements in the array.
211 * Print the result to the console.
212 */
213
214 int sumOfNameLengths = 0;
215 for (int num : nameLengths) {
216     sumOfNameLengths += num;
217 }
218 System.out.println("Requirement 5:");
219 System.out.println("-----");
220 System.out.print("The elements in the nameLengths array are: ");
221 for (int i=0; i<nameLengths.length; i++) {
222     System.out.print(nameLengths[i] + " ");
223 }
224 System.out.println();
225 System.out.println();
226
227 /*
228 * Write a method that takes a String, word and an int, n,
229 * as arguments and returns the word concatenated to itself
230 * n number of times.(i.e. if I pass in "Hello" and 3, I would
231 * expect the method to return "HelloHelloHello").
232 */
233 System.out.println("Requirement 6:");
234 System.out.println("-----");
235 System.out.print("The sum of all of the elements "
236     + "in the nameLengths array is: " + sumOfNameLengths);
237 System.out.println();
238 System.out.println();
239
240 String inputString = "Hello";
241 int inputNum = 3;
242 System.out.println("Input variables: " + inputString + " & " + inputNum);
243 System.out.println("Output of method multiplyConcat is: " + multiplyConcat (inputString, inputNum));
244 System.out.println();
245
246 inputString = "Goodbye";
247 inputNum = 5;
248 System.out.println("Input variables: " + inputString + " & " + inputNum);
249 System.out.println("Output of method multiplyConcat is: " + multiplyConcat (inputString, inputNum));
250 System.out.println();
251 System.out.println();
252
253 /*
254 * Write a method that takes two Strings, firstName and last Name,
255 * and returns a full name (the full name should be the first and
256 * last name as a String separated by a space).
257 */
258
259 String firstName = "Mickey";
260 String lastName = "Mouse";
261
262 System.out.println("Requirement 8:");
263 System.out.println("-----");
264 System.out.println("The first name is: " + firstName);
265 System.out.println("The last name is: " + lastName);
266 System.out.println("Output of method createFullName is: " + createFullName (firstName, lastName));
267 System.out.println();
268 System.out.println();
269
270 /*
271 * Write a method that takes an array of int and returns true if
272 * the sum of all the ints in the array is greater than 100.
273 */
274
275 int[] numArray = {60,3,7,25,88,71,3};
276 int[] numNewArray = {1,2,3,4,5,6,7,8,9,10};
277
278 System.out.println("Requirement 9:");
279 System.out.println("-----");
280
281 System.out.print("The elements in numArray are: ");
282 for (int i = 0; i<numArray.length; i++) {
283     System.out.print(numArray[i]+ " ");
284 }
285 System.out.println();
286
287 System.out.print("Does the sum of elements in numArray equal more than 100? ");
288 if (isGreaterThanOneHundred(numArray)) {
289     System.out.println("Yes");
290 } else {
291     System.out.println("No");
292 }
293 System.out.println();
294
295 System.out.print("The elements in numNewArray are: ");
296 for (int i = 0; i<numNewArray.length; i++) {
297     System.out.print(numNewArray[i]+ " ");
298 }
299 System.out.println();
300 System.out.print("Does the sum of elements in numNewArray equal more than 100? ");
301 if (isGreaterThanOneHundred(numNewArray)) {
302     System.out.println("Yes");
303 } else {
304     System.out.println("No");
305 }
306 System.out.println();
307 System.out.println();
308
309 /*
310 * Write a method that takes an array of double &
311 * returns the average of all the elements in the array.
312 */
313
314 double[] pricesOfItems = {2.00, 4.00, 3.45, 7.75, 3.25, 9.99};
315 System.out.println("Requirement 10:");
316 System.out.println("-----");
317 System.out.print("The elements in the pricesOfItems array are: ");
318 for (int i=0; i<pricesOfItems.length; i++) {
319     System.out.printf("%.2f", pricesOfItems[i]);
320 }
321 System.out.println();
322 System.out.print("The average price of items in the pricesOfItems array is: $");
323 System.out.printf("%.2f", averageDouble(pricesOfItems));
324 System.out.println();
325 System.out.println();
326 System.out.println();
327 System.out.println();
328
329 /*
330 * Requirement 11
331 * Write a method that takes two arrays of double &
332 * returns true if the average of the elements in the
333 * first array is greater than the average of the elements
334 * in the second array.
335 */
336 double[] pricesOfNewBooks = {15.00,65.00,34.00,53.00,22.00};
337 double[] pricesOfUsedBooks = {14.99,54.99,32.50,17.32,15.46};
338

```

```

 328
 329     System.out.println("Requirement 11:");
 330     System.out.println("-----");
 331     System.out.print("The elements in the pricesOfNewBooks array are: ");
 332     for (int i=0; i<pricesOfNewBooks.length; i++) {
 333         System.out.printf("%.2f ", pricesOfNewBooks[i]);
 334     }
 335     System.out.println();
 336     System.out.println("-----");
 337     System.out.print("The elements in the pricesOfUsedBooks array are: ");
 338     for (int i=0; i<pricesOfUsedBooks.length; i++) {
 339         System.out.printf("%.2f ", pricesOfUsedBooks[i]);
 340     }
 341     System.out.println();
 342     System.out.println("-----");
 343     System.out.print("Is the average of the elements in the priceOfNewBooks array greater than ");
 344     System.out.print("the average of the elements in the priceOfUsedBooks array? ");
 345     if (isAverageGreaterThan(pricesOfNewBooks,pricesOfUsedBooks)) {
 346         System.out.println("Yes");
 347     } else {
 348         System.out.println("No");
 349     }
 350     System.out.println();
 351     System.out.println("-----");
 352     System.out.println();
 353     System.out.println("-----");
 354
 355
 356     System.out.println("Reversing the input to the method, isAverageGreaterThan:");
 357     System.out.println("-----");
 358     System.out.print("Is the average of the elements in the priceOfUsedBooks array greater than ");
 359     System.out.print("the average of the elements in the priceOfNewBooks array? ");
 360     if (isAverageGreaterThan(pricesOfUsedBooks,pricesOfNewBooks)) {
 361         System.out.println("Yes");
 362     } else {
 363         System.out.println("No");
 364     }
 365     System.out.println();
 366     System.out.println("-----");
 367     System.out.print("Is the average of the elements in the priceOfUsedBooks array greater than ");
 368     System.out.print("the average of the elements in the priceOfNewBooks array? ");
 369     if (isAverageGreaterThan(pricesOfUsedBooks,pricesOfNewBooks)) {
 370         System.out.println("Yes");
 371     } else {
 372         System.out.println("No");
 373     }
 374     System.out.println();
 375     System.out.println();
 376
 377
 378     /*
 379      * Requirement 12
 380      * Write a method called willBuyDrink that takes a boolean
 381      * isHotOutside and a double moneyInPocket, and returns true
 382      * if it is hot outside and if moneyInPocket is greater
 383      * than 10.50.
 384      */
 385     boolean isHotOutside = true;
 386     double moneyInPocket = 10.00;
 387     System.out.println("Requirement 12:");
 388     System.out.println("-----");
 389     System.out.print("You will only buy a drink if it is hot outside");
 390     System.out.println("and you have more than $10.50 in your pocket!");
 391     System.out.println("-----");
 392     System.out.print("Is it hot outside? ");
 393     if (isHotOutside) {
 394         System.out.println("Yes");
 395     } else {
 396         System.out.println("No");
 397     }
 398     System.out.print("How much money do you have in your pocket? $");
 399     System.out.print("%.2f",moneyInPocket);
 400     System.out.println("-----");
 401     System.out.print("Will you buy a drink? ");
 402     if (willBuyDrink(isHotOutside, moneyInPocket)) {
 403         System.out.println("Yes");
 404     } else {
 405         System.out.println("It must be hot outside, and you have more than $10.50!");
 406     }
 407     System.out.println("No");
 408     System.out.println();
 409     System.out.println();
 410
 411
 412     /*
 413      * Requirement 13
 414      * Create a method of your own that solves a problem.
 415      * In comments, write what the method does and why you created it.
 416      *
 417      * My set of methods is for a Point Of Sale System.
 418      * Methods:
 419      *     boolean pointOfSale (double purchasePrice, double moneyTendered)
 420      *         This method takes the total purchase price and the money tendered,
 421      *         and returns a boolean to indicate whether the sale was successful.
 422      *     double returnCorrectChange(double price, double moneyTendered)
 423      *         This method returns the difference between the price of an item
 424      *         and the money tendered, returning a message if not enough money
 425      *         has been tendered for the purchase of the item.
 426      *     double addItemPrices (double[], priceArray)
 427      *         This method returns the sum of all of the elements in a double array.
 428      */
 429
 430     System.out.println("Requirement 13:");
 431     System.out.println("-----");
 432
 433     /*
 434      * Example for Requirement 13 --- a simple first example
 435      */
 436
 437     int numberOfPurchase = 1;
 438     System.out.println("Purchase #" + numberOfPurchase + ":");
 439     System.out.println("-----");
 440     double priceOfItem = 14.55;
 441     double moneyTendered = 20.00;
 442     if (pointOfSale (priceOfItem, moneyTendered)) {
 443         System.out.println("Have a nice day!");
 444     }
 445     System.out.println();
 446     System.out.println();
 447
 448     /*
 449      * Example for Requirement 13 --- showing moneyTendered < priceOfItem
 450      */
 451     numberOfPurchase++;
 452     System.out.println("Purchase #" + numberOfPurchase + ":");
 453     System.out.println("-----");
 454     priceOfItem = 9.55;
 455     moneyTendered = 5.00;
 456     if (pointOfSale (priceOfItem, moneyTendered)) {
 457         System.out.println("Have a nice day!");
 458     }
 459     System.out.println();
 460     System.out.println();
 461
 462     /*
 463      * Example for Requirement 13 --- using a item price of $0.00
 464      */
 465     numberOfPurchase++;
 466     System.out.println("Purchase #" + numberOfPurchase + ":");
 467     System.out.println("-----");
 468     priceOfItem = 0.00;
 469     moneyTendered = 20.00;
 470     if (pointOfSale (priceOfItem, moneyTendered)) {
 471         System.out.println("Have a nice day!");
 472     }
 473     System.out.println();
 474     System.out.println();
 475
 476     /*
 477      * Example for Requirement 13 --- using an array of prices
 478      *             (need to call addItemPrices)
 479      */
 480     numberOfPurchase++;
 481     System.out.println("Purchase #" + numberOfPurchase + ":");
 482     System.out.println("-----");
 483
 484     System.out.println("Using an array of prices, and adding all of the items in the array!");
 485     System.out.println("-----");
 486     double[] itemPrices = {4.00, 6.45, 9.55, 10.50, 8.99};
 487     moneyTendered = 50.00;
 488
 489     System.out.println("The number of elements in the itemPrices array: " + itemPrices.length);
 490     System.out.print("The elements in the itemPrices array are: ");
 491     for (int i = 0; i<itemPrices.length; i++) {
 492         System.out.printf("%.2f ", itemPrices[i]);
 493     }
 494     System.out.println();
 495
 496  */

```

```

495
496     if (pointOfSale ((priceOfItem = addItemPrices(itemPrices)), moneyTendered)) {
497         System.out.println("Have a nice day!");
498     }
499     System.out.println();
500     System.out.println();
501
502     System.out.println("-----");
503     System.out.println("End of Intro To Java Week 3 Coding Assignment");
504     System.out.println("-----");
505 }
506
507 /**
508 * Methods Written for Requirements in Week 3
509 * ****
510 */
511
512 /**
513 * Requirement 7 -- multiplyConcat Method
514 */
515 public static String multiplyConcat (String word, int n) {
516     String concatenatedString = "";
517     for (int i = 0; i<n; i++) {
518         concatenatedString += word;
519     }
520     return concatenatedString;
521 }
522
523 /**
524 * Requirement 8 -- createFullName Method
525 */
526 public static String createFullName(String first, String last) {
527     return (first + " " + last);
528 }
529
530 /**
531 * Requirement 9 -- isGreaterThanOrEqualToOneHundred Method
532 */
533 public static boolean isGreaterThanOrEqualToOneHundred(int[] newarray) {
534     int sum = 0;
535     for (int num : newarray) {
536         sum += num;
537     }
538     if (sum > 100) {
539         return true;
540     } else {
541         return false;
542     }
543 }
544
545 /**
546 * Requirement 10 -- averageDouble Method
547 */
548 public static double averageDouble (double[] doubleArray) {
549     double sum = 0.0;
550     for (double num : doubleArray ) {
551         sum += num;
552     }
553     return sum/doubleArray.length;
554 }
555
556 /**
557 * Requirement 11 -- isAverageGreater Than Method
558 */
559 public static boolean isAverageGreater Than (double[] arrayOne, double[] arrayTwo) {
560     if (averageDouble(arrayOne) > averageDouble(arrayTwo)) {
561         return true;
562     } else {
563         return false;
564     }
565 }
566
567 /**
568 * Requirement 12 -- willBuyDrink Method
569 */
570 public static boolean willBuyDrink (boolean isHot, double hasMoney) {
571     if (isHot && hasMoney > 10.50) {
572         return true;
573     } else {
574         return false;
575     }
576 }
577
578 */
579 /**
580 * Requirement 13 -- A method of my own design.
581 * This ended up being a set of methods.
582 *
583 * My set of methods is for a Point Of Sale System.
584 * Methods:
585 *     boolean pointOfSale (double purchasePrice, double moneyTendered)
586 *         This method takes the total purchase price and the money tendered,
587 *         and returns a boolean to indicate whether the sale was successful.
588 *     double returnCorrectChange (double price, double moneyTendered)
589 *         This method returns the difference between the price of an item
590 *         and the money tendered, returning a message if not enough money
591 *         has been tendered for the purchase of the item.
592 *     double addItemPrices (double[] priceAarray)
593 *         This method returns the sum of all of the elements in a double array.
594 *     returnCorrectChange
595 *         This method takes two doubles as input,
596 *         the first is the price, the second is the money paid; and
597 *         returns a double, which is the difference of these two, or will
598 *         be set to the money paid amount if it is less than the price.
599 *         This method will return a negative value (meaning the amount paid)
600 *         will not be enough to purchase the item.
601 *         I created this method because it seems like any application used
602 *         for point of sale would need a method with this functionality.
603 *         It will also be useful for my example ice cream ordering program.
604 */
605
606 /**
607 * addItemPrices -- Adds all elements in an array of double
608 */
609 public static double addItemPrices (double[] itemArray) {
610     double sum = 0;
611     for (double item : itemArray) {
612         sum += item;
613     }
614     return sum;
615 }
616
617 /**
618 * pointOfSale -- Returns true if the sale is successful
619 */
620 public static boolean pointOfSale (double price, double money) {
621     double change = 0;
622     System.out.printf("The purchase price is: $");
623     System.out.print(price);
624     System.out.println();
625     System.out.printf("Your money tendered is: $");
626     System.out.print(money);
627     System.out.println();
628     if (price == money) {
629         System.out.println("This item is free!");
630         return true;
631     } else if (change = returnCorrectChange(price,money)) < money) {
632         System.out.printf("Your change is: $%.2f", change);
633         System.out.println();
634         return true;
635     } else {
636         System.out.println("NO SALE!");
637     }
638 }
639
640 */

```

## Screenshots of Running Application:

App.java    Console

<terminated> App [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0\_261.jdk/Contents/Home/bin/java (Oct 29, 2020, 7:02:01 PM)

Intro To Java Week 3 Coding Assignment

---

Requirement 1:  
Elements in the ages array are: 3 9 23 64 2 8 28 93

Requirement 1(a):  
The first array element is: 3  
The last array element is: 93  
The difference in the ages array between  
the last array element & the first array element is: 90

Requirement 1(b):  
Elements in the ages2 array are: 3 9 23 64 2 8 28 93 100  
The first array element is: 3  
The last array element is: 100  
The difference in the ages2 array (e.g. the ages array plus one element)  
between the last array element & the first array element is: 97

Requirement 1(c):  
The average of all of the elements in the ages array is: 28  
The average of all of the elements in the ages2 array is: 36

Requirement 2:  
The length of the names array is: 6  
The names in the names array are: Sam Tommy Tim Sally Buck Bob

Requirement 2(a):  
The average number of letters per name in the names array is: 3

Requirement 2(b):  
The concatenation of the elements in the names array with spaces is: Sam Tommy Tim Sally Buck Bob

Requirement 3:  
Using our declared arrays, ages & names; and  
the index <arrayName>.length-1

The last element of the ages array is: 93  
The last element of the names array is: Bob

Requirement 4:  
Using our declared arrays, ages & names;  
and the index <arrayName>[0]

The first element of the ages array is: 3  
The first element of the names array is: Sam

Requirement 5:  
The elements in the nameLengths array are: 3 5 3 5 4 3

Requirement 6:  
The sum of all of the elements in the nameLengths array is: 23

Requirement 7:  
The following method takes input of a String, word & an int,  
n, and returns the word concatenated to itself n times.

---

Input variables: Hello & 3  
Output of method multiplyConcat is: HelloHelloHello

Input variables: Goodbye & 5  
Output of method multiplyConcat is: GoodbyeGoodbyeGoodbyeGoodbyeGoodbye

Requirement 8:  
The first name is: Mickey  
The last name is: Mouse  
Output of method createFullName is: Mickey Mouse

Requirement 9:  
The elements in numArray are: 60 3 7 25 88 71 3  
Does the sum of elements in numArray equal more than 100? Yes

The elements in numNewArray are: 1 2 3 4 5 6 7 8 9 10  
Does the sum of elements in numNewArray equal more than 100? No

Requirement 10:  
The elements in the pricesOfItems array are: 2.00 4.00 3.45 7.75 3.25 9.99  
The average price of items in the pricesOfItems array is: \$5.07

Requirement 11:  
The elements in the pricesOfNewBooks array are: 15.00 65.00 34.00 53.00 22.00  
The elements in the pricesOfUsedBooks array are: 14.99 54.99 32.50 17.32 15.46  
Is the average of the elements in the priceOfNewBooks array greater than  
the average of the elements in the priceOfUsedBooks array? Yes

Reversing the input to the method, isAverageGreaterThan:

Is the average of the elements in the priceOfUsedBooks array greater than  
the average of the elements in the priceOfNewBooks array? No

Requirement 12:

You will only buy a drink if it is hot outside  
and you have more than \$10.50 in your pocket!

Is it hot outside? Yes

How much money do you have in your pocket? \$10.00

Will you buy a drink? No

Requirement 13:

Purchase #1:

The purchase price is: \$14.55

The money tendered is: \$20.00

Your change is: \$5.45

Have a nice day!

Purchase #2:

The purchase price is: \$9.55

The money tendered is: \$5.00

NO SALE: More money is needed to make this purchase!

NO SALE!

Purchase #3:

The purchase price is: \$0.00

The money tendered is: \$20.00

This item is free!

Have a nice day!

Purchase #4:

Using an array of prices, and adding all of the items in the array!

The number of elements in the itemPrices array: 5

The elements in the itemPrices array are: 4.00 6.45 9.55 10.50 8.99

The purchase price is: \$39.49

The money tendered is: \$50.00

Your change is: \$10.51

Have a nice day!

End of Intro To Java Week 3 Coding Assignment

## URL to GitHub Repository:

<https://github.com/sw-dev-lisa-s-nh/IntroToJava-week3>