

Intro to Java Week 4 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 instance of an ArrayList of String called employeeNames
2. Create an instance of a HashSet of Integer called ids
3. Create an instance of a HashMap of Integer, String called employeeMap
4. Add at least five entries to the employeeNames and ids (make sure both collections have the same number of entries).
5. Create a variable **int i = 0;** then iterate over ids using an enhanced for loop. Inside the enhanced for loop use **employeeMap.put()** to add a new entry to the map. The entry should consist of a key that is the id in the enhanced loop's current iteration, and a value

that is the employeeName at position **i** of the employeeNames ArrayList. Increment **i** so that each iteration grabs the next element in the ArrayList.

6. Once the employeeMap is fully populated, use another enhanced for loop to iterate over the **employeeMap.keySet()**, and use the key for each current iteration to print to the console both the current key and its associated value in the map.
7. Create a StringBuilder called idsBuilder.
8. Iterate over the ids HashSet and append each id, followed by a dash "-" to idsBuilder.
9. Print the result of **idsBuilder.toString()** to the console.
10. Create another StringBuilder called namesBuilder.
11. Iterate over the employeeNames ArrayList and append each name, followed by a space " " to the namesBuilder.
12. Print the result of **namesBuilder.toString()** to the console.

Screenshots of Code:

```
Week4App.java  Console
1 package com.lisasmith.week4;
2
3 import java.util.ArrayList;
4 import java.util.HashMap;
5 import java.util.HashSet;
6 import java.util.List;
7 import java.util.Map;
8 import java.util.Set;
9
10 public class Week4App {
11
12     public static void main(String[] args) {
13
14         /*
15          * Requirement 1 — Create an instance of an ArrayList of String
16          * called employeeNames
17          */
18         List<String> employeeNames = new ArrayList<String>();
19
20         /*
21          * Requirement 2 — Create an instance of a HashSet of Integer called jds
22          */
23         Set<Integer> ids = new HashSet<Integer>();
24
25         /*
26          * Requirement 3 — Create an instance of a HashMap of Integer, String
27          * called employeeMap
28          */
29         Map<Integer, String> employeeMap = new HashMap<Integer, String>();
30
31         /*
32          * Requirement 4 — Add at least five entries to the employeeNames and jds
33          * (make sure they each have the same number of entries)
34          */
35         employeeNames.add("George Washington");
36         employeeNames.add("John Adams");
37         employeeNames.add("Thomas Jefferson");
38         employeeNames.add("James Madison");
39         employeeNames.add("James Monroe");
40         employeeNames.add("John Adams");
41         employeeNames.add("Andrew Jackson");
42         employeeNames.add("Martin Van Buren");
43         employeeNames.add("William Harrison");
44         employeeNames.add("John Tyler");
45
46         ids.add(101);
47         ids.add(102);
48         ids.add(103);
49         ids.add(104);
50         ids.add(105);
51         ids.add(106);
52         ids.add(107);
53         ids.add(108);
54         ids.add(109);
55         ids.add(110);
56
57         System.out.println("-----");
58         System.out.println("IntroToJava-Week4 Coding Assignment");
59         System.out.println("-----");
60         System.out.println();
61
62         System.out.println("Requirement 1, 2, & 4");
63         System.out.println();
64
65         System.out.println("Employee Names:");
66         System.out.println("-----");
67         for (String name : employeeNames) {
68             System.out.println(name);
69         }
70         System.out.println();
71
72         System.out.println("ids:");
73         System.out.println("-----");
74         for (Integer id : ids) {
75             System.out.println(id);
76         }
77         System.out.println();
78
79         /*
80          * Requirement 5
81          * Create a variable jds i = 0; then iterate over jds
82          * using an enhanced for loop. Inside the enhanced for
83          * loop use employeeMap.put() to add a new entry to the map.
84          * The entry should consist of a key that is the id in the
85          * enhanced loop's current iteration, and a value that is
86          * the employeeName at position i of the employeeNames ArrayList.
87          * Increment i so that each iteration grabs the next element
88          * in the ArrayList.
89          */
90         int i = 0;
91         for (Integer id : ids) {
92             employeeMap.put(id, employeeNames.get(i++));
93         }
94
95         /*
96          * Requirement 6 — Once the employeeMap is fully populated, use another
97          * enhanced for loop to iterate over the employeeMap.keySet(), and use
98          * the key for each current iteration to print to the console both the
99          * current key and it's associated value in the map.
100          */
101         System.out.println("Requirements 3, 5, & 6");
102         System.out.println("-----");
103         System.out.println();
104
105         System.out.println("Employee Map Contents");
106         System.out.println("-----");
107         System.out.println("Key: Value:");
108         System.out.println("-----");
109         Set<Integer> employeeKeys = employeeMap.keySet();
110         for (Integer key : employeeKeys) {
111             System.out.println(key + " : " + employeeMap.get(key));
112         }
113         System.out.println();
114
115         /*
116          * Requirement 7 — Create a StringBuilder called idsBuilder.
117          */
118         StringBuilder idsBuilder = new StringBuilder();
119
120         /*
121          * Requirement 8 — Iterate over the jds HashSet and append
122          * each id, followed by a dash "-" to
123          * idsBuilder.
124          */
125         for (Integer id : ids) {
126             idsBuilder.append(id + "-");
127         }
128
129         /*
130          * Requirement 9 — print the result of idsBuilder.toString() to the console.
131          */
132         System.out.println("Requirements 7-9:");
133         System.out.println("-----");
134         System.out.println();
135         System.out.println(idsBuilder.toString());
136         System.out.println();
137         System.out.println();
138
139         /*
140          * Requirement 10 — Create another StringBuilder called namesBuilder
141          */
142         StringBuilder namesBuilder = new StringBuilder();
143
144         /*
145          * Requirement 11 — Iterate over the employeeNames ArrayList and
146          * append each name, followed by a space " " to
147          * the namesBuilder.
148          */
149         for (String name : employeeNames) {
150             namesBuilder.append(name + " ");
151         }
152
153         /*
154          * Requirement 12 — Print the result of namesBuilder.toString() to the console.
155          */
156         System.out.println("Requirements 10-12:");
157         System.out.println("-----");
158         System.out.println();
159         System.out.println(namesBuilder.toString());
160         System.out.println();
161         System.out.println();
162
163         System.out.println("-----");
164         System.out.println("End of IntroToJava-Week4 Coding Assignment");
165         System.out.println("-----");
166     }
167 }
```

Screenshots of Running Application:

```
Week4App.java Console
<terminated> Week4App [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_261.jdk/Contents/Home/bin/java (Nov 6, 2020, 6:18:59 PM – 6:18:59 PM)

IntroToJava-Week4 Coding Assignment
-----

Requirement 1, 2, & 4:
-----

Employee Names:
-----
George Washington
John Adams
Thomas Jefferson
James Madison
James Monroe
John Adams
Andrew Jackson
Martin Van Buren
William Harrison
John Tyler

Ids:
-----
101
102
103
104
105
106
107
108
109
110

Requirements 3, 5, & 6:
-----

Employee Map Contents
-----
Key: Value:
-----
101 : George Washington
102 : John Adams
103 : Thomas Jefferson
104 : James Madison
105 : James Monroe
106 : John Adams
107 : Andrew Jackson
108 : Martin Van Buren
109 : William Harrison
110 : John Tyler

Requirements 7-9:
-----

101-102-103-104-105-106-107-108-109-110-

Requirements 10-12:
-----

George Washington John Adams Thomas Jefferson James Madison James Monroe John Adams Andrew Jackson Martin Van Buren William Harrison John Tyler

-----
End of IntroToJava-Week4 Coding Assignment
-----
```

URL to GitHub Repository:

<https://github.com/sw-dev-lisa-s-nh/IntroToJava-week4.git>