

Intro to Java Week 5 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 interface named Logger.
2. Add two void methods to the Logger interface, each should take a String as an argument
 - a. Log
 - b. Error
3. Create two classes that implement the Logger interface
 - a. AsteriskLogger
 - b. SpacedLogger

4. The log method on the AsteriskLogger should print out the String it receives between 3 asterisks on either side of the String (e.g. if the String passed in is “Hello”, then it should print ***Hello*** to the console.
5. The error method on the AsteriskLogger should print the String it receives inside a box of asterisks, with the String preceded by the word “ERROR:”. For example, if “Hello” is the argument, the following should be printed:

```
*****
```

```
***Error: Hello***
```

```
*****
```

6. The SpacedLogger should add spaces between each character of the String argument passed into its methods.
7. If the log method received “Hello” as an argument, it should print H e l l o
8. The error method should do the same, but with “ERROR:” preceding the spaced out input (i.e. ERROR: H e l l o)
9. Create a class named App that has a main method.
10. In this class instantiate an instance of each of your logger classes that implement the Logger interface.
11. Test both methods on both instances, passing in Strings of your choice.

Screenshots of Code:

```
App.java  Logger.java  AsteriskLogger.java  SpacedLogger.java  Cons

1 package com.lisasmith.week5;
2
3 import java.util.Date;
4
5 public class App {
6
7     // static Logger logger;
8
9     public static void main(String[] args) {
10
11         /*
12          * Requirement 9: Create a class named App that has a main method!
13          */
14
15         System.out.println("-----");
16         System.out.println("Intro To Java -- Week 5");
17         System.out.println("-----");
18         System.out.println();
19
20         /*
21          * Requirement 10: In this class, instantiate an instance
22          * of each of your logger classes that implements
23          * the Logger interface
24          */
25
26         Logger asteriskLogger = new AsteriskLogger();
27         Logger spacedLogger = new SpacedLogger();
28
29         /*
30          * Requirement 11: Test both methods (Log & Error) on both instances
31          * (asteriskLogger & spacedLogger), with Strings of
32          * my choice.
33          */
34         * (1) Initial Test Case: "Hello"
35         * (2) Longer String (more than 5 characters): "Buenos Dias"
36         * (3) Actual Log Event: using Date
37         * (4) Actual Error Event: Test .Error with an Error Message
38         */
39
40         /*
41          * (1) Initial Test case -- "Hello"
42          */
43
44         System.out.println("Requirement 11:");
45         System.out.println("-----");
46         System.out.println();
47
48         System.out.println();
49         System.out.println("(1) Initial test using \"Hello\"");
50         System.out.println();
51
52         System.out.println("AsteriskLogger:");
53         System.out.println("-----");
54         System.out.println();
55         asteriskLogger.Log("Hello");
56         System.out.println();
57         asteriskLogger.Log("Hello");
58         System.out.println();
59         asteriskLogger.Error("Hello");
60         System.out.println();
61
62         System.out.println("SpacedLogger:");
63         System.out.println("-----");
64         System.out.println();
65         spacedLogger.Log("Hello");
66         System.out.println();
67         spacedLogger.Log("Hello");
68         System.out.println();
69         spacedLogger.Error("Hello");
70         System.out.println();
71
72         /*
73          * (2) Longer String -- Notice that the asterisks are printed
74          * to match the length of the string.
75          */
76
77         System.out.println();
78         System.out.println("(2) Test using a longer string");
79         System.out.println("-----");
80
81         System.out.println("AsteriskLogger:");
82         System.out.println("-----");
83         System.out.println();
84         asteriskLogger.Log("Buenos Dias");
85         System.out.println();
86         asteriskLogger.Log("Buenos Dias");
87         System.out.println();
88         asteriskLogger.Error("Buenos Dias");
89         System.out.println();
90
91         System.out.println("SpacedLogger:");
92         System.out.println("-----");
93         System.out.println();
94         spacedLogger.Log("Buenos Dias");
95         System.out.println();
96         spacedLogger.Log("Buenos Dias");
97         System.out.println();
98         spacedLogger.Error("Buenos Dias");
99         System.out.println();
100
101         /*
102          * (3) Actual Log Event -- Notice that the asterisks are printed
103          * to match the length of the string.
104          */
105
106         System.out.println();
107         System.out.println("(3) Test using the Date");
108         System.out.println("-----");
109
110         Date date = new Date();
111
112         ...
113
114         System.out.println("AsteriskLogger:");
115         System.out.println("-----");
116         System.out.println();
117         asteriskLogger.Log(date.toString());
118         System.out.println();
119         asteriskLogger.Error(date.toString());
120         System.out.println();
121
122         System.out.println("SpacedLogger:");
123         System.out.println("-----");
124         System.out.println();
125         spacedLogger.Log(date.toString());
126         System.out.println();
127         spacedLogger.Error(date.toString());
128         System.out.println();
129
130         /*
131          * (4) Actual Error Event -- Notice that the asterisks are printed
132          * to match the length of the string.
133          */
134
135         System.out.println();
136         System.out.println("(4) Test .Error Methods using an error message");
137         System.out.println("-----");
138
139         /*
140          * Variables created for (4) Test -- sum is never used
141          */
142         int[] digits = {0,1,2,3,4,5,6,7,8,9};
143         int index = 0;
144         int sum = 0;
145
146         /*
147          * This is a contrived error scenario -- I intentionally tried to access one index past the
148          * last element of an array to force the try/catch to catch the exception.
149          * This is programmer's error, but done intentionally. I wasn't sure how to get a test
150          * case with a "real" exception, so I forced this by putting a index <= digits.length.
151          * (NOTE: to be coded correctly this should be index<digits.length)
152          */
153
154         /*
155          * Contrived example: sum of all of the entries in the digits array.
156          * This gives us a test case using an actual exception.
157          */
158         System.out.println("AsteriskLogger:");
159         System.out.println("-----");
160         System.out.println();
161         System.out.println();
162         System.out.println();
163
164         while (index<=digits.length) {
165             try {
166                 sum = digits[index++];
167             } catch (Exception e){
168                 asteriskLogger.Error(e.toString());
169             }
170         }
171         System.out.println();
172         System.out.println();
173
174         /*
175          * Contrived example: sum of all of the entries in the digits array.
176          * This gives us a test case using an actual exception.
177          */
178         System.out.println("SpacedLogger:");
179         System.out.println("-----");
180         System.out.println();
181         System.out.println();
182         System.out.println();
183         index = 0;
184         sum = 0;
185
186         while (index<=digits.length) {
187             try {
188                 sum = digits[index++];
189             } catch (Exception e){
190                 spacedLogger.Error(e.toString());
191             }
192         }
193
194         System.out.println();
195         System.out.println();
196
197         System.out.println("-----");
198         System.out.println("End of Intro To Java -- Week 5");
199         System.out.println("-----");
200
201     }
202
203 }
204
205 }
206
207 }
```

```
App.java | Logger.java | AsteriskLogger.java | SpacedLogger.java | Console
1 package com.lisasmith.week5;
2
3 /*
4  * Requirement 1: Create an interface named Logger.
5  */
6
7 public interface Logger {
8
9     /*
10      * Requirement 2a: Create void method Log,
11      *                  which takes a String as an argument.
12      */
13     public void Log(String log);
14
15     /*
16      * Requirement 2b: Create void method Error,
17      *                  which takes a String as an argument.
18      */
19     public void Error(String error);
20
21 }
22
```

```
App.java | Logger.java | AsteriskLogger.java | SpacedLogger.java | Console
1 package com.lisasmith.week5;
2
3
4 /*
5  * Requirement 3a: Create the Asterisk Logger class that implements the Logger interface
6  */
7
8 public class AsteriskLogger implements Logger {
9
10
11
12     /*
13      * Requirement 4:
14      * The log method on the AsteriskLogger should print out the String it receives
15      * between 3 asterisks on either side of the String.
16      * (e.g. if the String passed in is "Hello", then it should print out
17      *      "***Hello***" to the console.
18      */
19
20     @Override
21     public void Log(String log) {
22         System.out.println("***" + log + "***");
23     }
24
25     /*
26      * Requirement 5:
27      * The error method on the AsteriskLogger should print out the String it receives
28      * inside a box of asterisks, with the String preceded by the word "ERROR:".
29      * For example: if "Hello" is the argument, the following should be printed:
30      *
31      * *****
32      * ***Hello***
33      * *****
34      */
35
36     @Override
37     public void Error(String error) {
38         String s1 = "***Error: ";
39         String s2 = "***";
40         int counter = s1.length() + s2.length() + error.length();
41         for (int i=0; i<counter; i++) {
42             System.err.print("*");
43         }
44         System.err.println();
45         System.err.println("***Error: " + error + "***");
46
47         for (int i=0; i<counter; i++) {
48             System.err.print("*");
49         }
50         System.err.println();
51     }
52 }
53
```

```
App.java | Logger.java | AsteriskLogger.java | SpacedLogger.java | Console
1 package com.lisasmith.week5;
2
3
4 /*
5  * Requirement 3b: Create the SpacedLogger class that implements the Logger interface
6  */
7
8 public class SpacedLogger implements Logger {
9
10
11
12     /*
13      * Requirement 6 & 7:
14      * Add spaces between each character of the String argument
15      * passed into the method. The log method should print the
16      * String passed in with a space between each character.
17      * (e.g. Hello --> "H e l l o")
18      */
19
20     @Override
21     public void Log(String log) {
22         for (int i = 0; i<log.length(); i++) {
23             System.out.print(log.charAt(i) + " ");
24         }
25         System.out.println();
26     }
27
28
29     /*
30      * Requirement 6 & 8:
31      * Add spaces between each character of the String argument
32      * passed into the method. The error method should add "ERROR: "
33      * preceding the spaced out input.
34      * (e.g. Hello --> "ERROR: H e l l o")
35      */
36
37
38     @Override
39     public void Error(String error) {
40         System.err.print("ERROR: ");
41         for (int i = 0; i<error.length(); i++) {
42             System.err.print(error.charAt(i) + " ");
43         }
44         System.err.println();
45     }
46
47 }
48
49
```

Screenshots of Running Application:



```
Console 22  Logger.java  AsteriskLogger.java  SpacedLogger.java  App.java
<terminated> App [Java Application] J:\Library\Java\JavaVirtualMachines\jdk1.8.0_261\jdk\Contents\Home\bin\java (Nov 10, 2020, 10:23)

Intro To Java -- Week 5

Requirement 11:
-----

(1) Initial test using "Hello"
-----
AsteriskLogger:
-----
***Hello***
*****
***Error: Hello***
*****
SpacedLogger:
-----
H e l l o

ERROR: H e l l o

(2) Test using a longer string
-----
AsteriskLogger:
-----
***Buenos Dias***
*****
***Error: Buenos Dias***
*****
SpacedLogger:
-----
B u e n o s   D i a s

ERROR: B u e n o s   D i a s

(3) Test using the Date
-----
AsteriskLogger:
-----
***Tue Nov 10 22:23:45 EST 2020***
*****
***Error: Tue Nov 10 22:23:45 EST 2020***
*****
SpacedLogger:
-----
T u e   N o v   1 0   2 2 : 2 3 : 4 5   E S T   2 0 2 0

ERROR: T u e   N o v   1 0   2 2 : 2 3 : 4 5   E S T   2 0 2 0

(4) Test .Error Methods using an error message
-----
AsteriskLogger:
-----
*****
***Error: java.lang.ArrayIndexOutOfBoundsException: 10***
*****
SpacedLogger:
-----
ERROR: j a v a . l a n g . A r r a y I n d e x O u t O f B o u n d s E x c e p t i o n :   1 0

End of Intro To Java -- Week 5
```

URL to GitHub Repository: <https://github.com/sw-dev-lisa-s-nh/IntroToJava-week5.git>