

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:

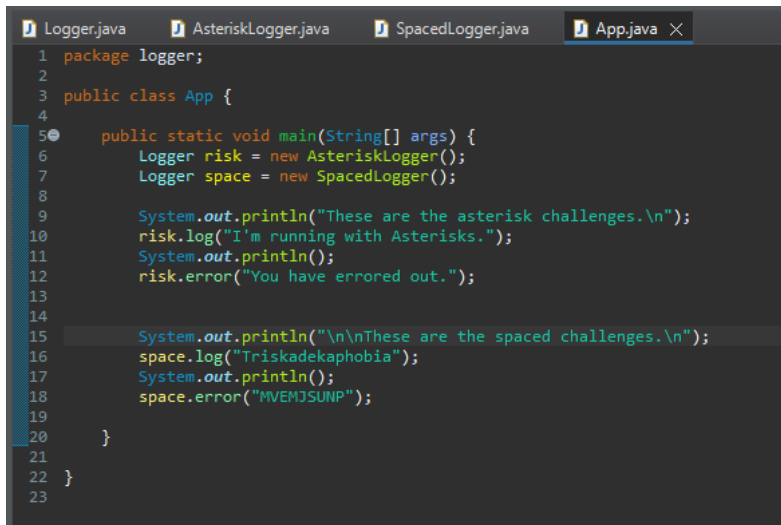
1. Create an interface named `Logger`.
2. Add two void methods to the `Logger` interface, each should take a `String` as an argument
 - a. `Log`
 - b. `Error`
3. Create two classes that implement the `Logger` interface
 - a. `AsteriskLogger`
 - b. `SpacedLogger`
4. The `log` method on the `AsteriskLogger` 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.

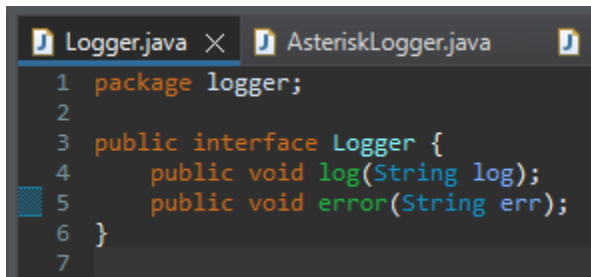
Screenshots of Code:

App.java:

A screenshot of an IDE showing the App.java file. The code defines a public class App with a main method. Inside the main method, two Logger objects are created: risk (AsteriskLogger) and space (SpacedLogger). The risk logger is used to print a message about asterisk challenges, and the space logger is used to print a message about spaced challenges. Both loggers are also used to print error messages.

```
1 package logger;
2
3 public class App {
4
5     public static void main(String[] args) {
6         Logger risk = new AsteriskLogger();
7         Logger space = new SpacedLogger();
8
9         System.out.println("These are the asterisk challenges.\n");
10        risk.log("I'm running with Asterisks.");
11        System.out.println();
12        risk.error("You have errored out.");
13
14
15        System.out.println("\n\nThese are the spaced challenges.\n");
16        space.log("Triskadekaphobia");
17        System.out.println();
18        space.error("MVEMJSUNP");
19    }
20 }
21
22 }
23
```

Logger.java:

A screenshot of an IDE showing the Logger.java file. The code defines a public interface Logger with two methods: log and error.

```
1 package logger;
2
3 public interface Logger {
4     public void log(String log);
5     public void error(String err);
6 }
7
```

AsteriskLogger:

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

1 package logger;
2
3 public class AsteriskLogger implements Logger {
4
5     /**
6      * log(String log)
7      * @param log
8      * Takes the string log and prints out
9      * the string: ***log***
10     */
11     public void log(String log) {
12         System.out.println("***" + log + "***");
13     }
14
15     /**
16      * error(String err)
17      * @param err
18      * Takes the string err and prints out
19      * three strings that run:
20      * *****
21      * ***Error: err***
22      * *****
23      * by repeating the * symbol as many
24      * times as the err string is long.
25     */
26     public void error(String err) {
27         /* saves to a variable so we can have an
28          * accurate count of how many stars we want. */
29         String builtString = "***Error: " + err + "***";
30
31         System.out.println("*".repeat(builtString.length()));
32         System.out.println(builtString);
33         System.out.println("*".repeat(builtString.length()));
34     }
35
36 }
```

SpacedLogger:

```
Logger.java  AsteriskLogger.java  SpacedLogger.java  App.java
1  package logger;
2
3  public class SpacedLogger implements Logger {
4
5      /**
6       * log(String log)
7       * @param log
8       * Takes the string log and prints out
9       * the string: l o g by passing the log String
10      * through the buildString function
11      */
12      public void log(String log) {
13          System.out.println(buildString(log));
14      }
15
16      /**
17       * error(String err)
18       * @param err
19       * Takes the string err and prints out
20       * three strings that run:
21       *     Error: e r r
22       * by passing the string err
23       * through the buildString function
24       */
25      public void error(String err) {
26          System.out.println("Error: " + buildString(err));
27      }
28      /**
29       *
30       * @param msg
31       * @return
32       */
33      public String buildString(String msg) {
34          String str = "";
35          for (int i = 0; i < msg.length(); i++) {
36              str = str + msg.charAt(i) + " ";
37          }
38          String formatString = str.trim();
39          return formatString;
40      }
41
42  }
43
```

Screenshots of Running Application:

```
These are the asterisk challenges.  
***I'm running with Asterisks.***  
  
*****  
***Error: You have errored out.***  
*****  
  
These are the spaced challenges.  
T r i s k a d e k a p h o b i a  
Error: M V E M J S U N P
```

URL to GitHub Repository:

<https://github.com/KitiaraJ/BootCamp/tree/main/java-wk5-challenge>