

Intro to Java Week 5 Coding Assignment

Tyler Livermore

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.

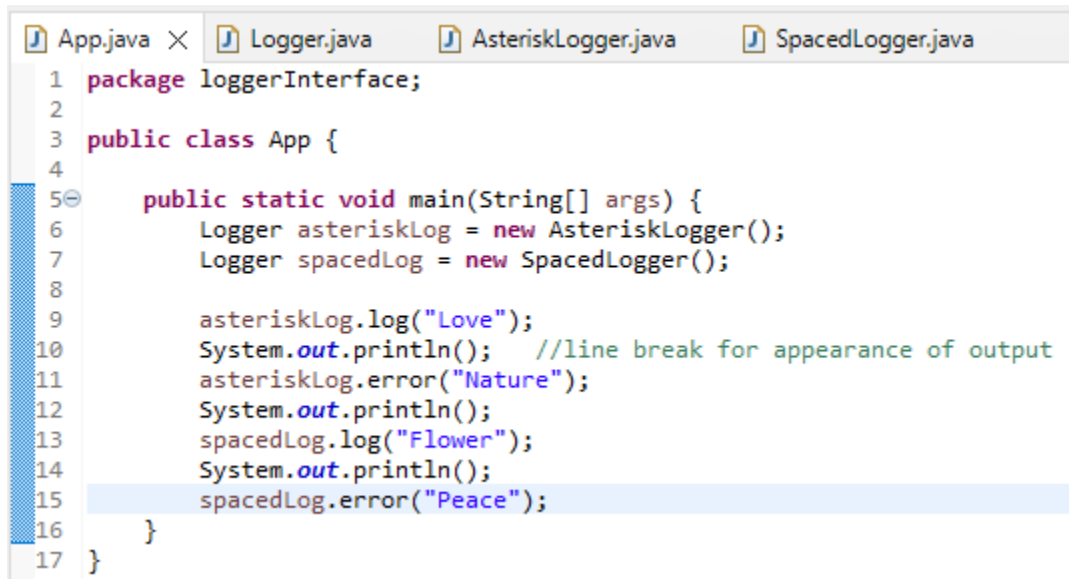
Coding Steps:

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:

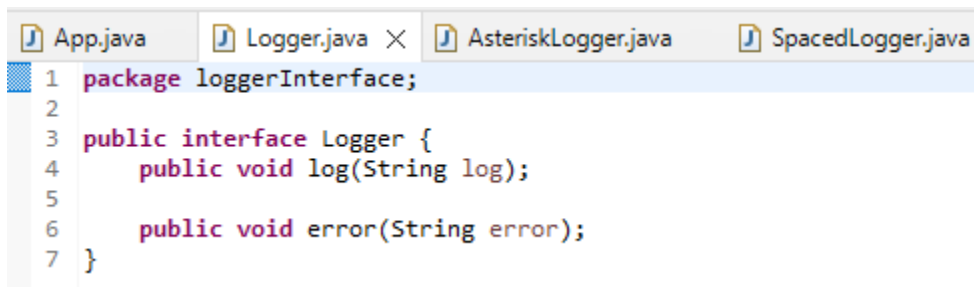
Error: Hello

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 X  Logger.java  AsteriskLogger.java  SpacedLogger.java
1 package loggerInterface;
2
3 public class App {
4
5     public static void main(String[] args) {
6         Logger asteriskLog = new AsteriskLogger();
7         Logger spacedLog = new SpacedLogger();
8
9         asteriskLog.log("Love");
10        System.out.println(); //line break for appearance of output
11        asteriskLog.error("Nature");
12        System.out.println();
13        spacedLog.log("Flower");
14        System.out.println();
15        spacedLog.error("Peace");
16    }
17 }
```

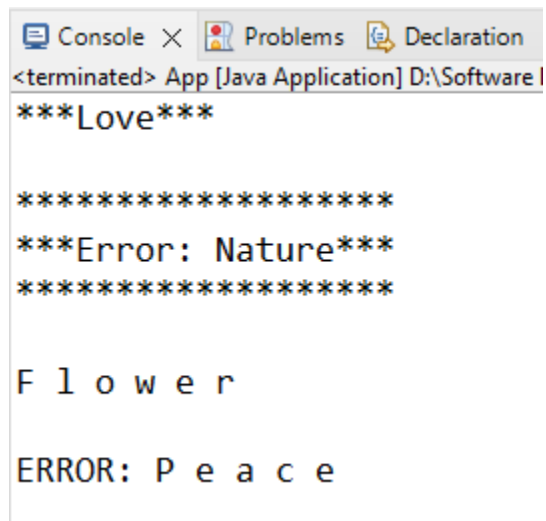


```
App.java  Logger.java X  AsteriskLogger.java  SpacedLogger.java
1 package loggerInterface;
2
3 public interface Logger {
4     public void log(String log);
5
6     public void error(String error);
7 }
```

```
App.java  Logger.java  AsteriskLogger.java  X  SpacedLogger.java
1 package loggerInterface;
2
3 public class AsteriskLogger implements Logger {
4     @Override
5     public void log(String log) {
6         //started with *** and appended on
7         StringBuilder messageLine = new StringBuilder("***");
8         messageLine.append(log + messageLine);
9
10        System.out.println(messageLine);
11    }
12    @Override
13    public void error(String error) {
14        int boxWidth = 0;
15        int i = 0;
16        StringBuilder messageLine = new StringBuilder("***");
17        StringBuilder asteriskBox = new StringBuilder("");
18
19        messageLine.append("Error: " + error + messageLine);
20
21        //Determine how long to make the top & bottom line asterisks to match the width of the user's string
22        boxWidth = messageLine.length();
23        while (i < boxWidth) {
24            asteriskBox.append("*");
25            i++;
26        }
27        System.out.println(asteriskBox + "\n" + messageLine + "\n" + asteriskBox);
28    }
29 }
```

```
App.java  Logger.java  AsteriskLogger.java  SpacedLogger.java  X
1 package loggerInterface;
2
3 public class SpacedLogger implements Logger {
4     @Override
5     public void log(String log) {
6         StringBuilder messageLine = new StringBuilder(log);
7         int length = messageLine.length();
8         int i = (length - 1); // -1 prevents creation of trailing space
9
10        //iterate down to 0 from the end of the string b/c
11        //iterating up causes an infinite loop since the string length increases with each space inserted
12        while (i > 0) {
13            messageLine.insert(i, ' ');
14            i--;
15        }
16        System.out.println(messageLine);
17    }
18    @Override
19    public void error(String error) {
20        StringBuilder messageLine = new StringBuilder(error);
21        StringBuilder header = new StringBuilder("ERROR: ");
22
23        int length = messageLine.length();
24        int i = (length - 1);
25
26        while (i > 0) {
27            messageLine.insert(i, ' ');
28            i--;
29        }
30        System.out.println(header.toString() + messageLine);
31    }
32 }
```

Screenshots of Running Application:



The screenshot shows an IDE console window with three tabs: 'Console', 'Problems', and 'Declaration'. The 'Console' tab is active, displaying the following text: '<terminated> App [Java Application] D:\Software I', '***Love***', a line of asterisks, '***Error: Nature***', another line of asterisks, 'F l o w e r', and 'ERROR: P e a c e'.

```
<terminated> App [Java Application] D:\Software I
***Love***

*****
***Error: Nature***
*****

F l o w e r

ERROR: P e a c e
```

URL to GitHub Repository:

https://github.com/tylerjlivermore/Week5_LoggerInterface