

# 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:

```
1 import java.util.*;
2
3 public interface Logger {
4     String w = "Hope";
5
6     public void Log();
7     public void Error();
8 }
9
10
11
```

```
1
2 public class AsteriskLogger implements Logger{
3
4     public void Log() {
5         System.out.println("****" + w + "****");
6     }
7     public void Error() {
8         System.out.println("*****");
9         System.out.println("***Error:" + w + "****");
10        System.out.println("*****");
11    }
12 }
13
```

```
1
2 public class SpacedLogger implements Logger{
3
4     public void Log() {
5         System.out.println(w.replaceAll("(?!$)", "$0 "));
6     }
7     public void Error() {
8         System.out.println("ERROR:" + w.replaceAll("(?!$)", "$0 "));
9     }
10 }
11
```

```
1 public class App {  
2  
3     public static void main(String[] args) {  
4  
5         AsteriskLogger astrisklogger = new AsteriskLogger();  
6         astrisklogger.Log();  
7         astrisklogger.Error();  
8  
9         SpacedLogger spacedlogger = new SpacedLogger();  
10        spacedlogger.Log();  
11        spacedlogger.Error();  
12    }  
13  
14 }
```

### Screenshots of Running Application:

```
***Hope***  
*****  
***Error:Hope***  
*****  
H o p e  
ERROR:H o p e
```

### URL to GitHub Repository:

<https://github.com/MThunder2/Week-5.git>