

# 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:

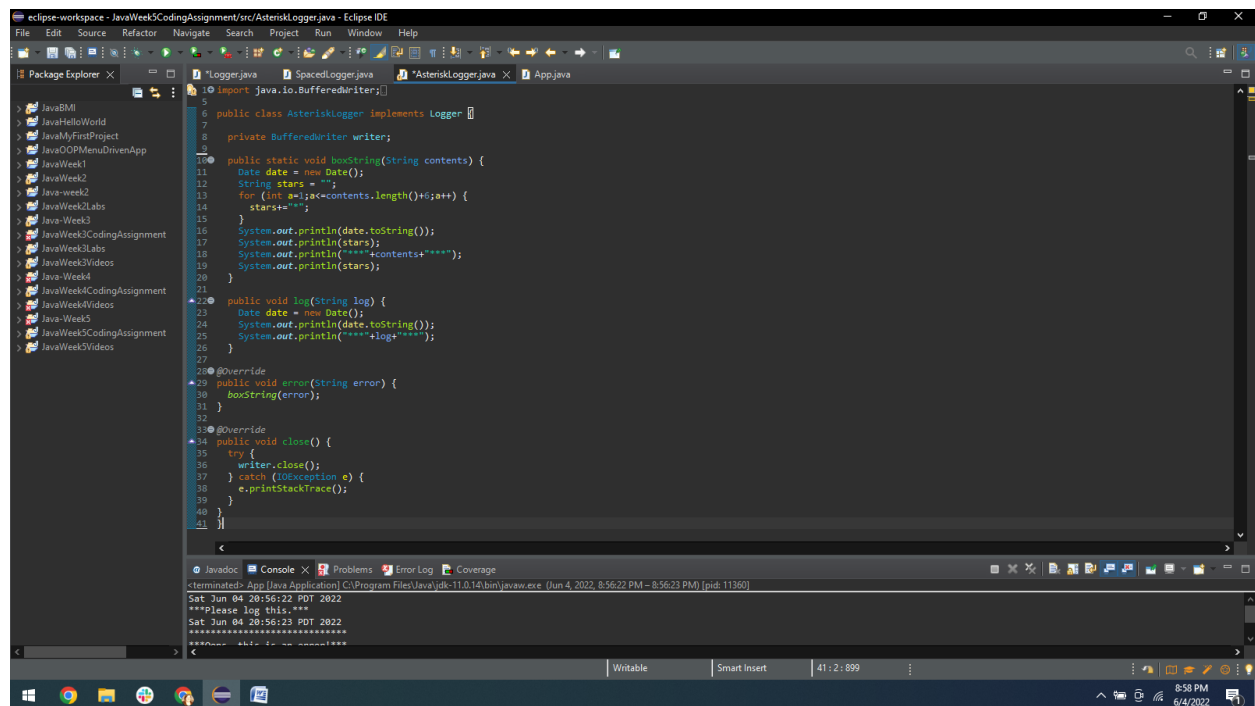
\*\*\*\*\*

\*\*\*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:



The screenshot shows the Eclipse IDE with the following code:

```
import java.io.BufferedWriter;

public class AsteriskLogger implements Logger {
    private BufferedWriter writer;

    public static void boxString(String contents) {
        Date date = new Date();
        String stars = "";
        for (int a=1;a<contents.length()+1;a++) {
            stars+=" ";
        }
        System.out.println(date.toString());
        System.out.println(stars);
        System.out.println("*****"+contents+"*****");
        System.out.println(stars);
    }

    public void log(String log) {
        Date date = new Date();
        System.out.println(date.toString());
        System.out.println("*****log*****");
    }

    @Override
    public void error(String error) {
        boxString(error);
    }

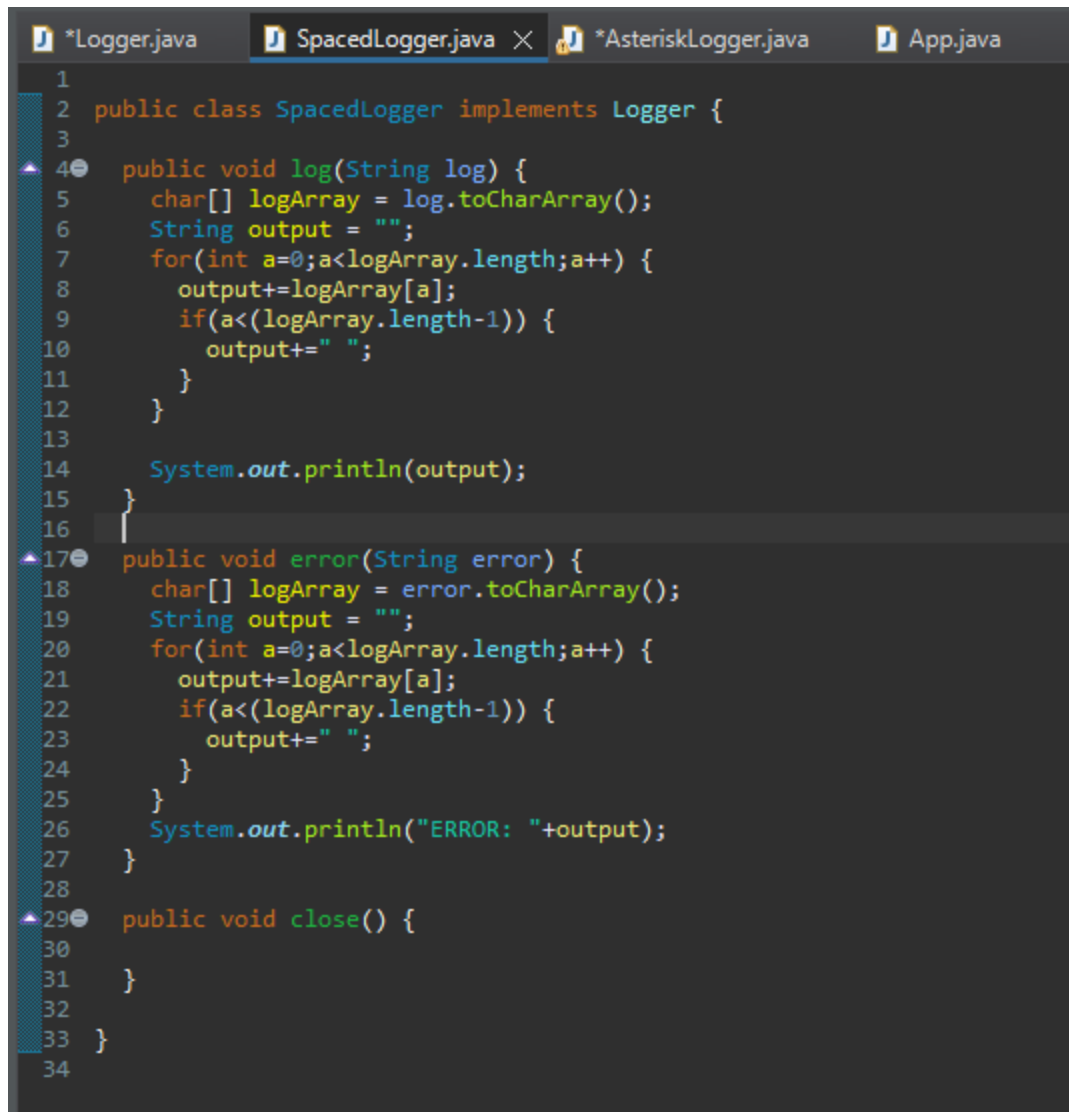
    @Override
    public void close() {
        try {
            writer.close();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

The console output shows the following:

```
terminated: App [Java Application] C:\Program Files\Java\jdk-11.0.14\bin\java.exe. (run 4, 2022, 8:56:22 PM - 8:56:23 PM) [pid: 11360]
Sat Jun 04 20:56:22 PDT 2022
****Please log this.***
Sat Jun 04 20:56:23 PDT 2022
*****
****Error: Hello****
*****
```

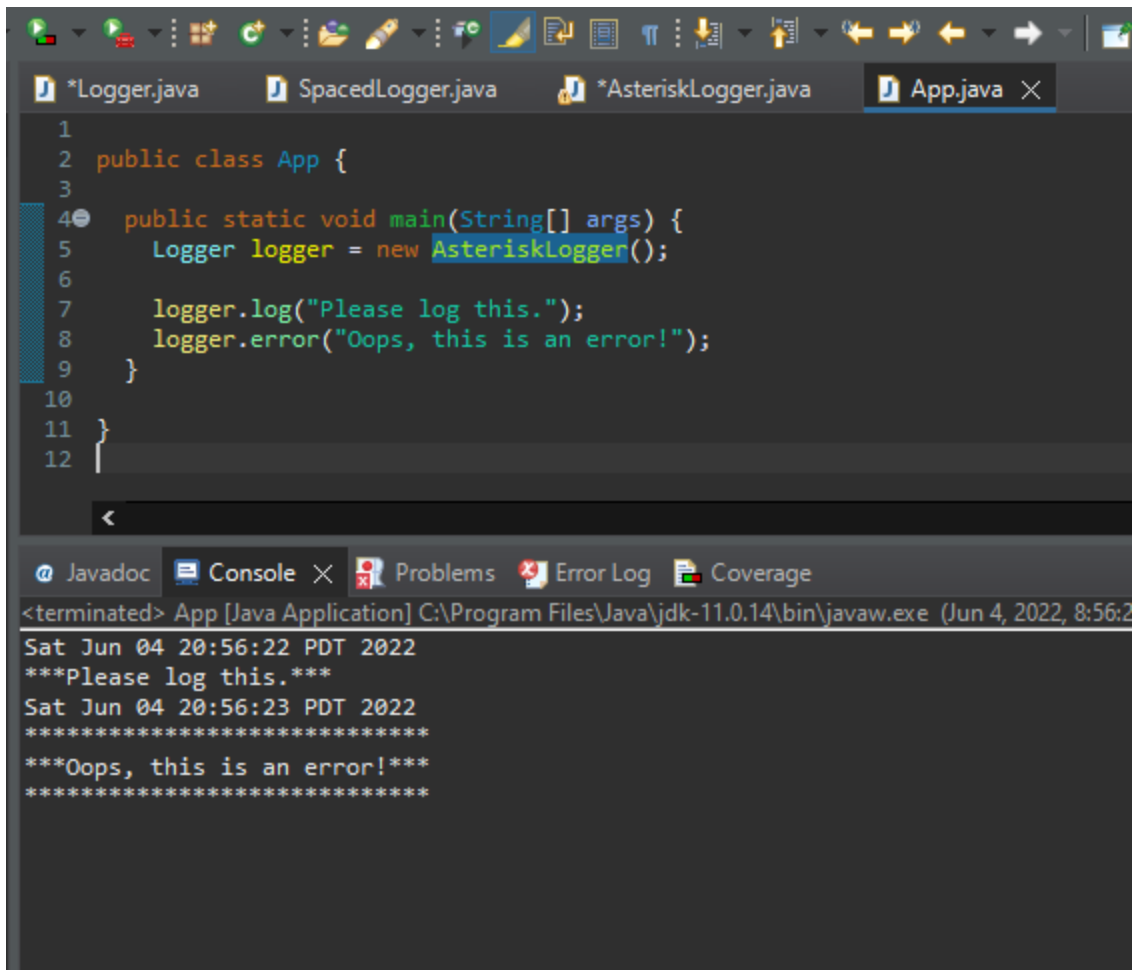
```
*Logger.java    SpacedLogger.java    *AsteriskLogger.java    App.java X
1
2 public class App {
3
4 public static void main(String[] args) {
5     Logger logger = new AsteriskLogger();
6
7     logger.log("Please log this.");
8     logger.error("Oops, this is an error!");
9 }
10
11 }
12
```

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



```
1
2 public class SpacedLogger implements Logger {
3
4     public void log(String log) {
5         char[] logArray = log.toCharArray();
6         String output = "";
7         for(int a=0;a<logArray.length;a++) {
8             output+=logArray[a];
9             if(a<(logArray.length-1)) {
10                 output+=" ";
11             }
12         }
13
14         System.out.println(output);
15     }
16
17     public void error(String error) {
18         char[] logArray = error.toCharArray();
19         String output = "";
20         for(int a=0;a<logArray.length;a++) {
21             output+=logArray[a];
22             if(a<(logArray.length-1)) {
23                 output+=" ";
24             }
25         }
26         System.out.println("ERROR: "+output);
27     }
28
29     public void close() {
30
31     }
32
33 }
34
```

**Screenshots of Running Application:**

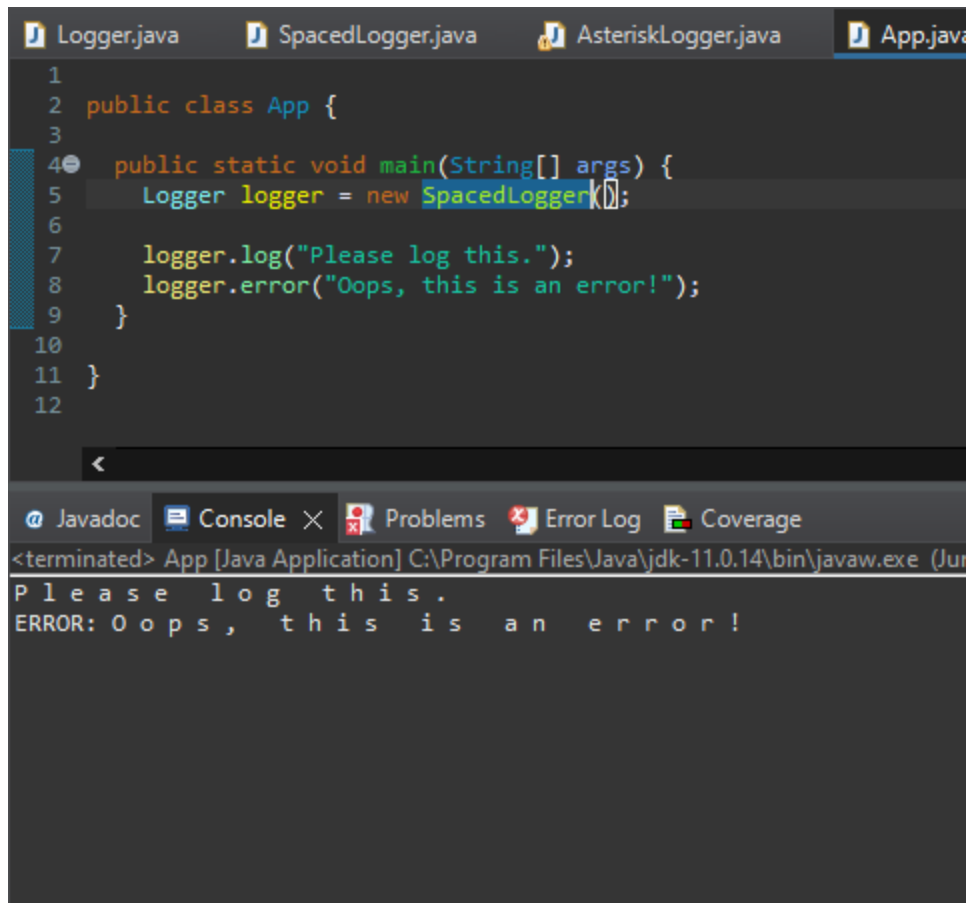


The screenshot shows an IDE with a toolbar at the top and a tab bar below it. The tabs are labeled `*Logger.java`, `SpacedLogger.java`, `*AsteriskLogger.java`, and `App.java`. The `App.java` tab is active, showing the following code:

```
1
2 public class App {
3
4     public static void main(String[] args) {
5         Logger logger = new AsteriskLogger();
6
7         logger.log("Please log this.");
8         logger.error("Oops, this is an error!");
9     }
10
11 }
12
```

Below the code editor is a panel with tabs for `Javadoc`, `Console`, `Problems`, `Error Log`, and `Coverage`. The `Console` tab is active, displaying the output of the application:

```
<terminated> App [Java Application] C:\Program Files\Java\jdk-11.0.14\bin\javaw.exe (Jun 4, 2022, 8:56:22)
Sat Jun 04 20:56:22 PDT 2022
***Please log this.***
Sat Jun 04 20:56:23 PDT 2022
*****
***Oops, this is an error!***
*****
```



The screenshot shows an IDE with four tabs: `Logger.java`, `SpacedLogger.java`, `AsteriskLogger.java`, and `App.java`. The `App.java` tab is active, displaying the following code:

```
1
2 public class App {
3
4     public static void main(String[] args) {
5         Logger logger = new SpacedLogger();
6
7         logger.log("Please log this.");
8         logger.error("Oops, this is an error!");
9     }
10
11 }
12
```

Below the code editor, there is a toolbar with icons for Javadoc, Console, Problems, Error Log, and Coverage. The Console tab is selected, showing the output of the program:

```
<terminated> App [Java Application] C:\Program Files\Java\jdk-11.0.14\bin\javaw.exe (J
Please log this .
ERROR: Oops, this is an error!
```

**URL to GitHub Repository:**

**<https://github.com/CodingVegas/JavaWeek5CodingAssignment>**