

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.

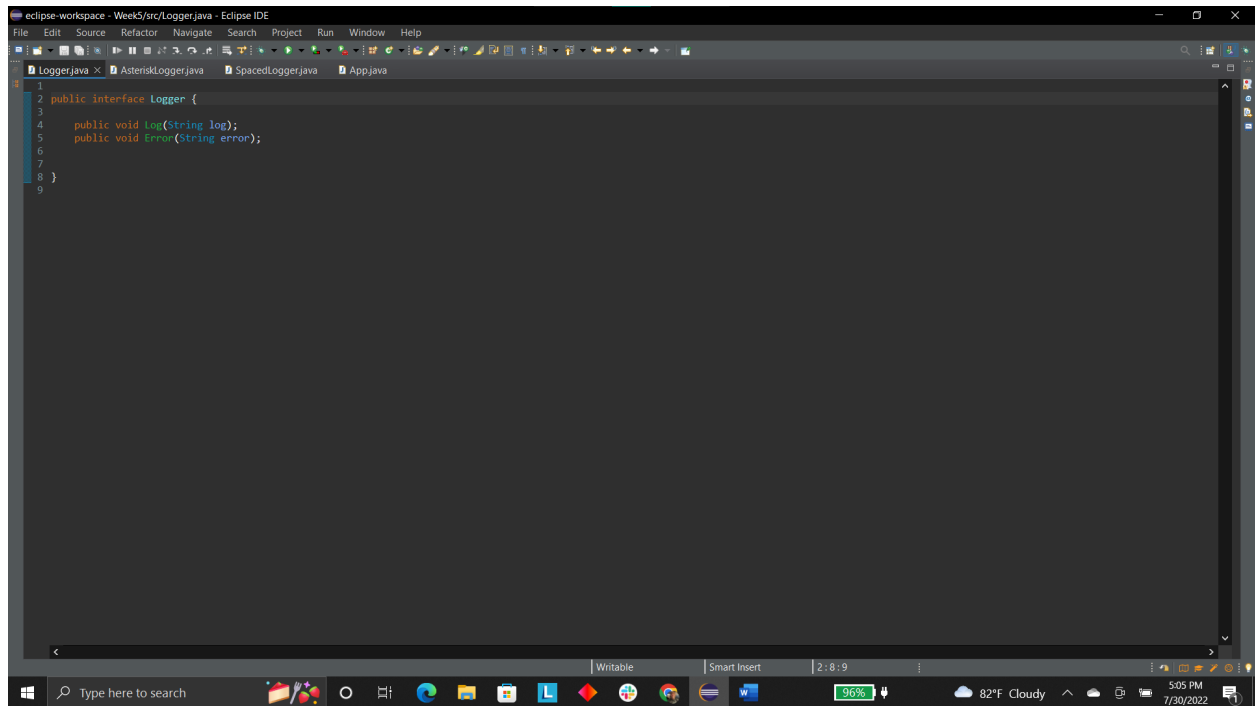
11. Test both methods on both instances, passing in Strings of your choice.

Screenshots of Code:

Finished Application:

```
1 import java.util.Scanner;
2
3 public class App {
4
5     static Scanner sc = new Scanner(System.in); //scanner to receive input
6
7
8
9
10    System.out.println("Please type any word!");
11    String word = sc.next();
12    System.out.println("");
13    System.out.println("Asterisk Logger function: ");
14    System.out.println("");
15
16    Logger aLogger = new AsteriskLogger();
17    aLogger.Log(word);
18    System.out.println("");
19    aLogger.Error(word);
20
21    System.out.println("//////////");
22    System.out.println("");
23    System.out.println("UnSpaced Logger function");
24    System.out.println("");
25
26    Logger sLogger = new SpacedLogger();
27    sLogger.Log(word);
28    System.out.println("");
29    sLogger.Error(word);
30
31    System.out.println("//////////");
32    System.out.println("Program over! Thanks!");
33 }
34
35
36 }
37
```

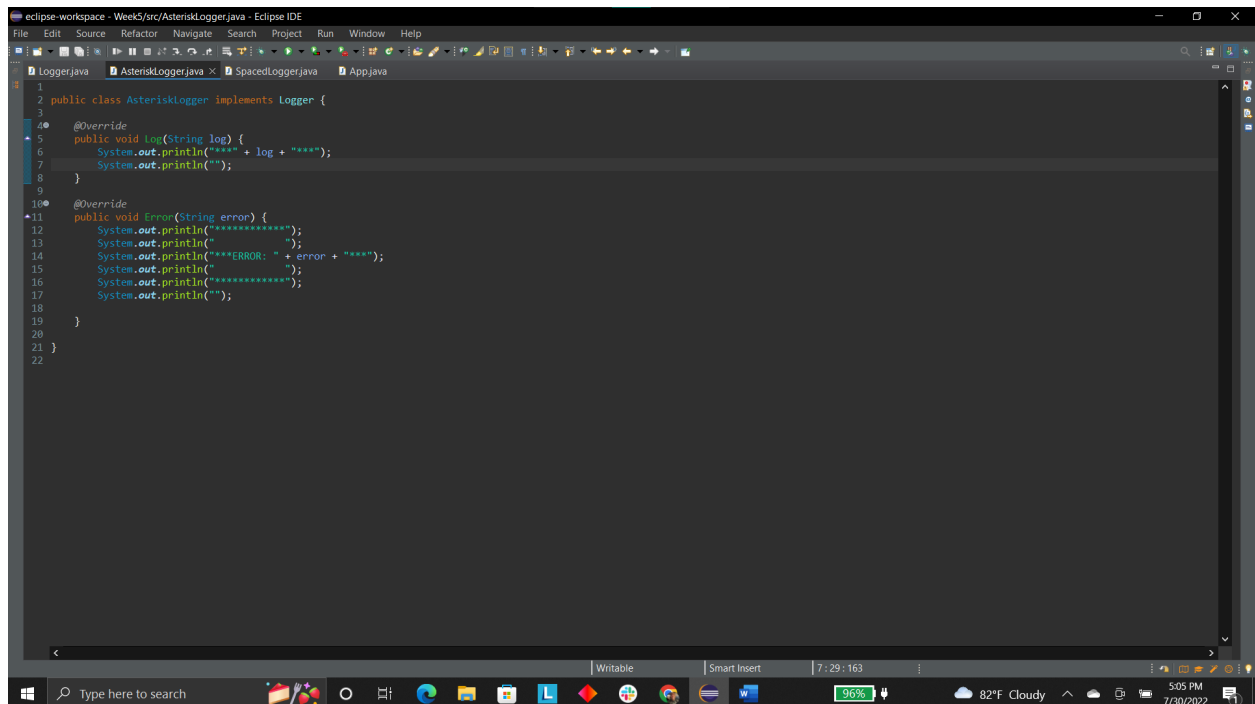
Logger Interface:



The screenshot shows the Eclipse IDE with the 'Logger.java' file open. The code defines a public interface named 'Logger' with two methods: 'log(String log)' and 'error(String error)'. The IDE's interface includes a menu bar (File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, Help), a toolbar with various icons, and a project explorer on the left showing the file structure. The status bar at the bottom indicates the file is 'Writable', 'Smart Insert' is active, and the system clock shows 5:05 PM on 7/30/2022.

```
1 public interface Logger {  
2  
3  
4     public void log(String log);  
5     public void error(String error);  
6  
7  
8 }  
9
```

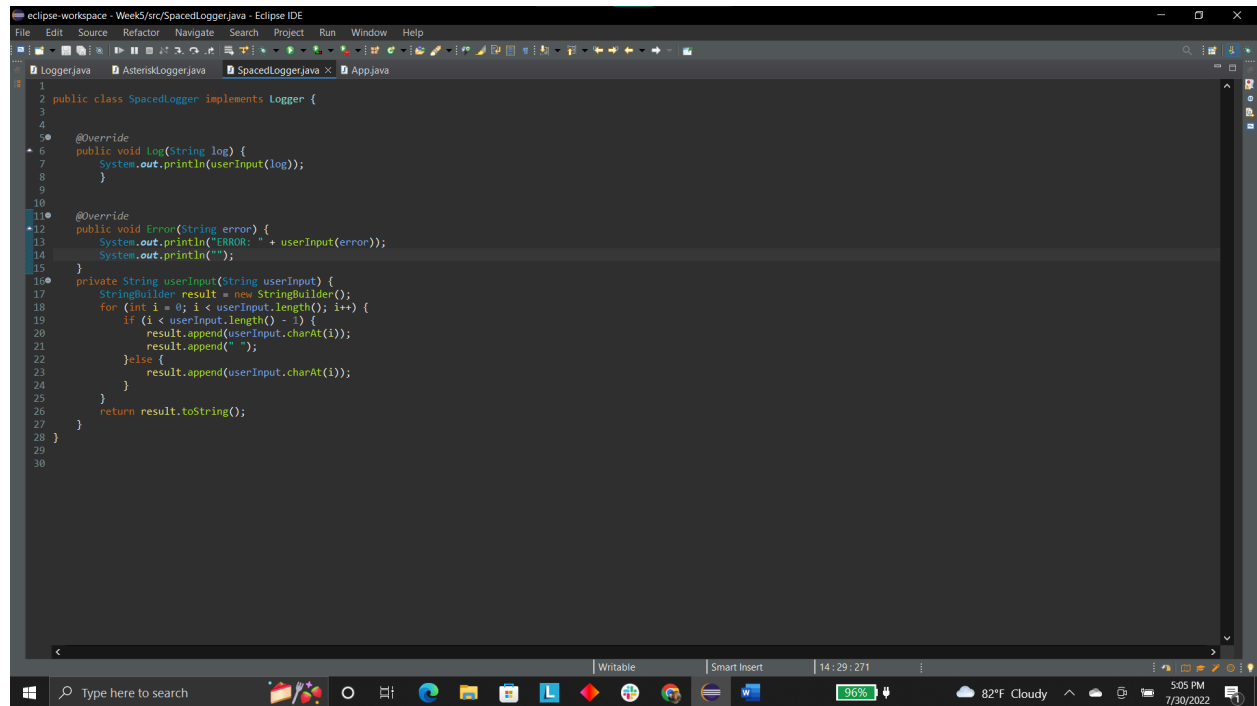
Asterisk Logger Class:



The screenshot shows the Eclipse IDE with the 'AsteriskLogger.java' file open. The code defines a public class 'AsteriskLogger' that implements the 'Logger' interface. It contains two methods: 'log(String log)' and 'error(String error)'. The 'log' method prints the log message with asterisks on either side. The 'error' method prints the error message with asterisks on either side and also prints 'ERROR: ' followed by the error message. The IDE's interface is the same as in the previous screenshot, showing the menu bar, toolbar, and project explorer. The status bar at the bottom indicates the file is 'Writable', 'Smart Insert' is active, and the system clock shows 5:05 PM on 7/30/2022.

```
1 public class AsteriskLogger implements Logger {  
2  
3  
4     @Override  
5     public void log(String log) {  
6         System.out.println("****" + log + "****");  
7         System.out.println("");  
8     }  
9  
10    @Override  
11    public void error(String error) {  
12        System.out.println("*****");  
13        System.out.println(" ");  
14        System.out.println("****ERROR: " + error + "****");  
15        System.out.println(" ");  
16        System.out.println("*****");  
17        System.out.println("");  
18    }  
19  
20 }  
21  
22
```

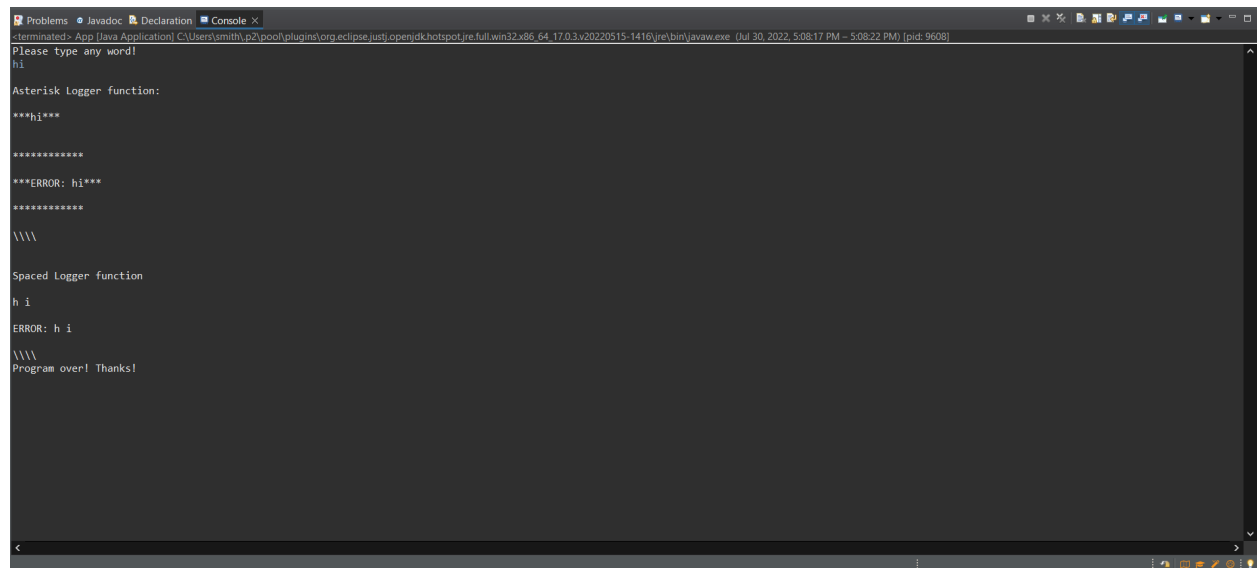
Spaced Logger Class:



The screenshot shows the Eclipse IDE with the file 'SpacedLogger.java' open. The code defines a 'SpacedLogger' class that implements a 'Logger' interface. It includes methods for logging, error handling, and a utility method for inserting spaces into a string.

```
1 public class SpacedLogger implements Logger {
2
3
4
5
6     @Override
7     public void log(String log) {
8         System.out.println(userInput(log));
9     }
10
11
12     @Override
13     public void Error(String error) {
14         System.out.println("ERROR: " + userInput(error));
15         System.out.println("");
16     }
17
18     private String userInput(String userInput) {
19         StringBuilder result = new StringBuilder();
20         for (int i = 0; i < userInput.length(); i++) {
21             if (i < userInput.length() - 1) {
22                 result.append(userInput.charAt(i));
23                 result.append(" ");
24             } else {
25                 result.append(userInput.charAt(i));
26             }
27         }
28         return result.toString();
29     }
30 }
```

Screenshot of Running Application:



The screenshot shows the Eclipse IDE console with the output of the 'App' application. The output demonstrates the 'Asterisk Logger function' and the 'Spaced Logger function'.

```
<terminated> App [Java Application] C:\Users\smith\p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64.17.0.3.v20220515-1416\jre\bin\java.exe (Jul 30, 2022, 5:08:17 PM - 5:08:22 PM) [pid: 9608]
Please type any word!
hi

Asterisk Logger function:
***hi***

*****
***ERROR: hi***
*****

\\

Spaced Logger function
h i

ERROR: h i

\\
Program over! Thanks!
```

URL to GitHub Repository:

[Repo](#)