

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.Scanner;
2
3 public class App {
4
5     static Scanner scanner = new Scanner(System.in);
6
7     public static void main(String[] args) {
8
9         String hello = "Hello";
10
11         System.out.println("Test of AsteriskLogger and SpacedLogger with short string: ");
12         presentAsterisk(hello);
13         presentSpaced(hello);
14
15         String longer = "This is a significantly longer String than hello";
16
17         System.out.println("Test of AsteriskLogger and SpacedLogger with short string: ");
18         presentAsterisk(longer);
19         presentSpaced(longer);
20
21
22
23         System.out.println("Now it's your turn:");
24
25         int decision = 0;
26         printOptions();
27         decision = scanner.nextInt();
28         if (decision == 1) {
29             System.out.println("Enter String without spaces to use: ");
30             String userStr = scanner.next();
31             presentAsterisk(userStr);
32         } else if (decision == 2) {
33             System.out.println("Enter String without spaces to use: ");
34             String userStr = scanner.next();
35             presentSpaced(userStr);
36         } else if (decision == 3) {
37             System.out.println("Enter String without spaces to use: ");
38             String userStr = scanner.next();
```

```

36         } else if (decision == 3) {
37             System.out.println("Enter String without spaces to use: ");
38             String userStr = scanner.next();
39             presentAsterisk(userStr);
40             presentSpaced(userStr);
41         } else {
42             System.out.println("Option not recognized, please restart program and choose valid option");
43         }
44         //Note that this is not a loop as this portion isn't required by the instructions but I wanted to add
45         //If I have time I will change this to be a loop similar to the menu app
46         //Additionally as scanner.nextLine() was causing problems with the if statements, it's just .next()
47         //I will also add the ability to use strings with spaces if I have time
48     }
49 }
50
51 private static void printOptions() {
52     System.out.println("1) Use AsteriskLogger with a new String");
53     System.out.println("2) Use SpacedLogger with a new String");
54     System.out.println("3) Use both with a new String");
55 }
56
57
58 public static void presentSpaced(String Str) {
59     SpacedLogger space = new SpacedLogger(Str);
60     space.Log();
61     space.Error();
62     System.out.println(" ");
63 }
64
65 public static void presentAsterisk(String Str) {
66     AsteriskLogger aster = new AsteriskLogger(Str);
67     aster.Log();
68     aster.Error();
69     System.out.println(" ");
70 }
71
72 }
73

```

```

1
2 public class AsteriskLogger implements Logger {
3
4     private String wordUsed;
5
6     public AsteriskLogger(String wordUsed) {
7         this.setWordUsed(wordUsed);
8     }
9
10    public void Log() {
11        System.out.println("****" + wordUsed + "****");
12    }
13
14
15    public void Error() {
16        for(int i = 0; i < (wordUsed.length() + 12); i++){
17            System.out.print("*");
18        }
19        System.out.println("");
20        System.out.println("****ERROR: " + wordUsed + "****");
21        for(int i = 0; i < (wordUsed.length() + 12); i++){
22            System.out.print("*");
23        }
24        System.out.println("");
25    }
26
27    public String getWordUsed() {
28        return wordUsed;
29    }
30
31    public void setWordUsed(String wordUsed) {
32        this.wordUsed = wordUsed;
33    }
34
35 }
36

```

```

1
2 public class SpacedLogger implements Logger {
3
4     private String wordUsed;
5
6     public SpacedLogger(String wordUsed) {
7         this.setWordUsed(wordUsed);
8     }
9
10    public void Log() {
11        for(int i = 0; i < wordUsed.length() - 1; i++) {
12            System.out.print(wordUsed.charAt(i) + " ");
13        }
14        System.out.println(wordUsed.charAt(wordUsed.length() - 1));
15    }
16
17    public void Error() {
18        System.out.print("ERROR: ");
19        for(int i = 0; i < wordUsed.length() - 1; i++) {
20            System.out.print(wordUsed.charAt(i) + " ");
21        }
22        System.out.println(wordUsed.charAt(wordUsed.length() - 1));
23    }
24
25    public String getWordUsed() {
26        return wordUsed;
27    }
28
29    public void setWordUsed(String wordUsed) {
30        this.wordUsed = wordUsed;
31    }
32
33 }
34
35
36

```

```

1
2 public interface Logger {
3     public void Log();
4     public void Error();
5 }
6

```

Screenshots of Running Application:

```

Test of AsteriskLogger and SpacedLogger with short string:
***Hello***
*****
***ERROR: Hello***
*****

H e l l o
ERROR: H e l l o

Test of AsteriskLogger and SpacedLogger with short string:
***This is a significantly longer String than hello***
*****
***ERROR: This is a significantly longer String than hello***
*****

T h i s   i s   a   s i g n i f i c a n t l y   l o n g e r   S t r i n g   t h a n   h e l l o
ERROR: T h i s   i s   a   s i g n i f i c a n t l y   l o n g e r   S t r i n g   t h a n   h e l l o

Now it's your turn:
1) Use AsteriskLogger with a new String
2) Use SpacedLogger with a new String
3) Use both with a new String
3
Enter String to use:
Valid
***Valid***
*****
***ERROR: Valid***
*****

V a l i d
ERROR: V a l i d

```

URL to GitHub Repository:

<https://github.com/kopatsis/Week5CodingAssignment>