

Intro to Java Week 5 Coding Assignment

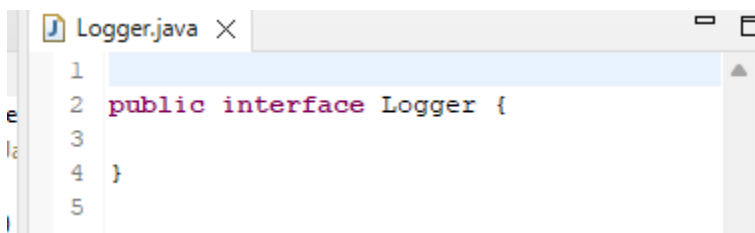
Points possible: 70

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.

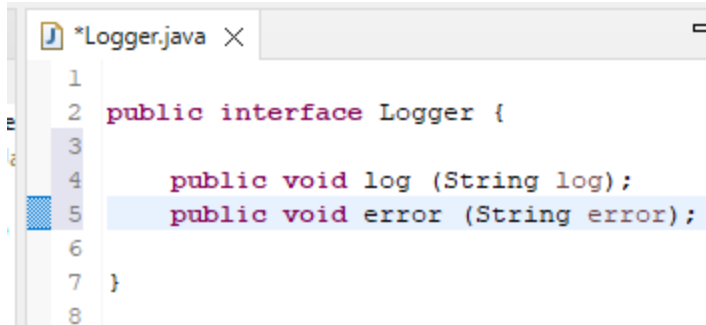
Coding Steps:

1. Create an interface named Logger.



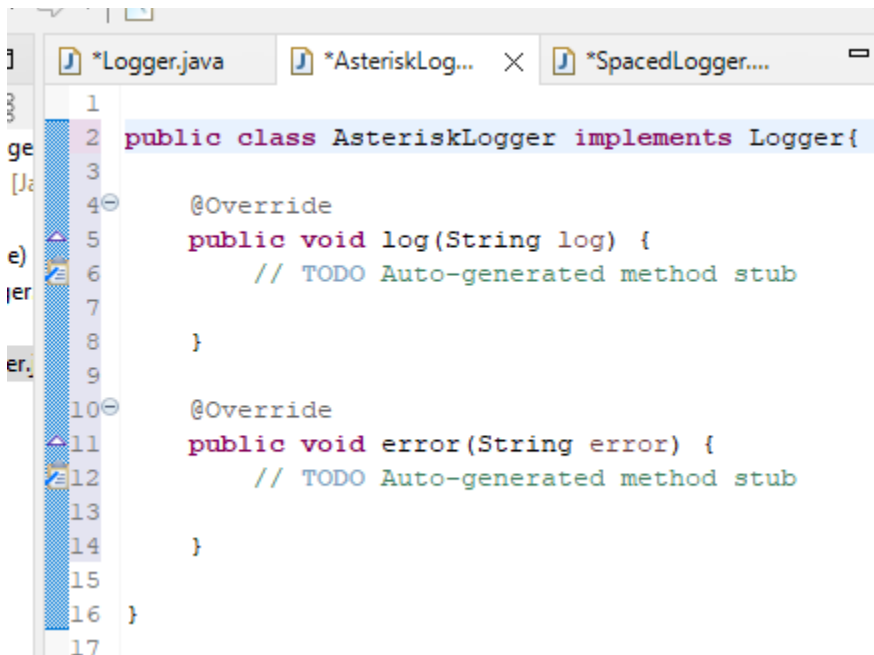
```
1 public interface Logger {
2
3
4 }
5
```

2. Add two void methods to the Logger interface, each should take a String as an argument
 - a. Log
 - b. Error



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

3. Create two classes that implement the Logger interface
 - a. AsteriskLogger



```
1
2 public class AsteriskLogger implements Logger{
3
4     @Override
5     public void log(String log) {
6         // TODO Auto-generated method stub
7
8     }
9
10    @Override
11    public void error(String error) {
12        // TODO Auto-generated method stub
13
14    }
15
16 }
17
```

- b. SpacedLogger

```

1
2 public class SpacedLogger implements Logger{
3
4     @Override
5     public void log(String log) {
6         // TODO Auto-generated method stub
7
8     }
9
10    @Override
11    public void error(String error) {
12        // TODO Auto-generated method stub
13
14    }
15
16 }
17

```

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.

```

@Override
public void log(String log) {
    System.out.println("*****");
    System.out.println("***" + log + "***");
    System.out.println("*****");
}

```

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:

```

-----
public void error(String error) {
    System.out.println("*****");
    System.out.println("***Error: " + error + "***");
    System.out.println("*****");
}

```

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

```

public void log(String log) {
    String spacedString = "";
    for (int i = 0 ; i < log.length(); i++) {
        spacedString = log.charAt(i)+ " ";
    }
    System.out.println("*****");
    System.out.println("****" + spacedString + "****");
    System.out.println("*****");
}

```

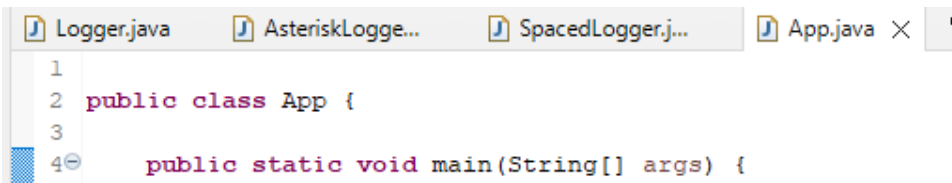
8. The error method should do the same, but with “ERROR:” preceding the spaced out input (i.e. ERROR: H e l l o)

```

public void error(String error) {
    String spacedString = "";
    for (int i = 0 ; i < error.length(); i++) {
        spacedString = error.charAt(i)+ " ";
    }
    System.out.println("*****");
    System.out.println("****Error: " + spacedString + "****");
    System.out.println("*****");
}

```

9. Create a class named App that has a main method.



```

1
2 public class App {
3
4     public static void main(String[] args) {

```

10. In this class instantiate an instance of each of your logger classes that implement the Logger interface.

```

1
2 public class App {
3
4     public static void main(String[] args) {
5         AsteriskLogger testAsteriskLogger = new AsteriskLogger();
6         SpacedLogger testSpacedLogger = new SpacedLogger();
7

```

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

Screenshots of Code:

Logger Interface:

```
Logger.java × AsteriskLogger.java SpacedLogger.java App.java
1
2 public interface Logger {
3
4     public void log (String log);
5     public void error (String error);
6
7 }
8
```

AsteriskLogger Class:

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

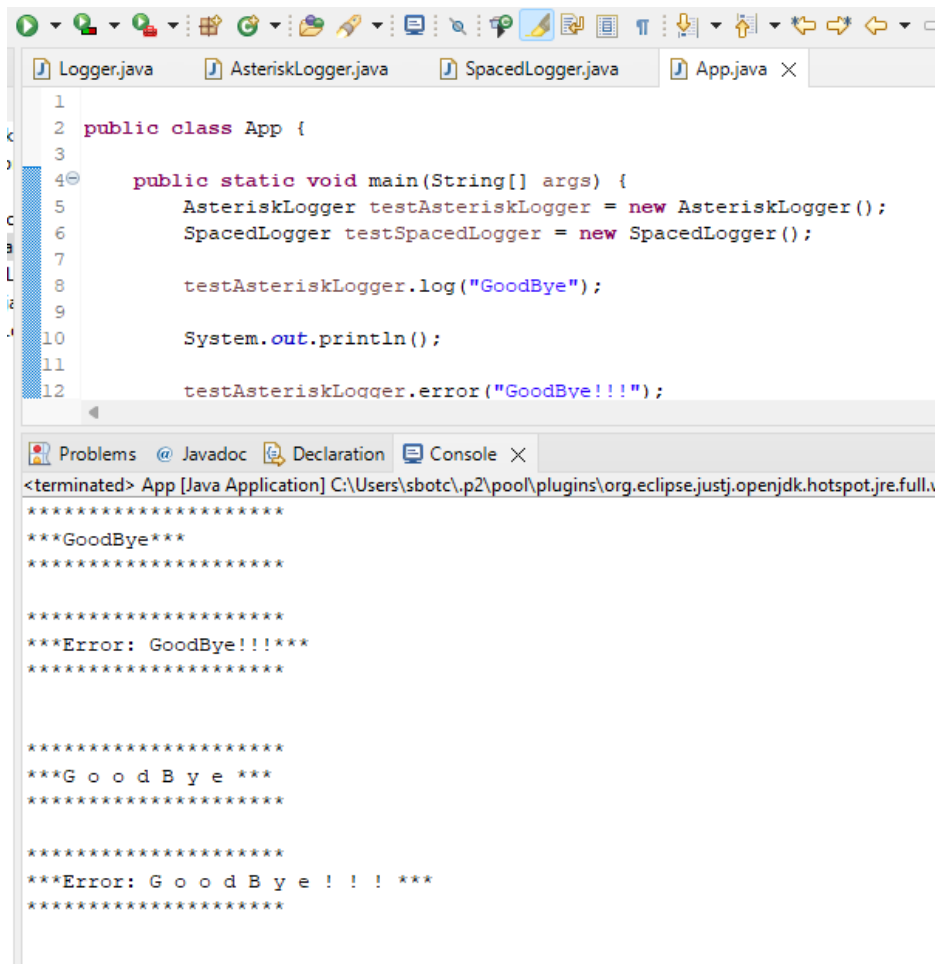
SpacedLogger Class:

```
Logger.java  AsteriskLogger.java  *SpacedLogger.java  App.java
1
2 public class SpacedLogger implements Logger{
3
4     @Override
5     public void log(String log) {
6         String spacedString = "";
7         for (int i = 0 ; i < log.length(); i++) {
8             spacedString += log.charAt(i)+ " ";
9         }
10        System.out.println("*****");
11        System.out.println("***" + spacedString + "***");
12        System.out.println("*****");
13    }
14
15    @Override
16    public void error(String error) {
17        String spacedString = "";
18        for (int i = 0 ; i < error.length(); i++) {
19            spacedString += error.charAt(i)+ " ";
20        }
21        System.out.println("*****");
22        System.out.println("***Error: " + spacedString + "***");
23        System.out.println("*****");
24    }
25 }
26
```

App Main Class:

```
Logger.java  AsteriskLogger.java  *SpacedLogger.java  App.java
1
2 public class App {
3
4     public static void main(String[] args) {
5         AsteriskLogger testAsteriskLogger = new AsteriskLogger();
6         SpacedLogger testSpacedLogger = new SpacedLogger();
7
8         testAsteriskLogger.log("GoodBye");
9
10        System.out.println();
11
12        testAsteriskLogger.error("GoodBye!!!");
13
14        System.out.println();
15        System.out.println();
16
17        testSpacedLogger.log("GoodBye");
18
19        System.out.println();
20
21        testSpacedLogger.error("GoodBye!!!");
22
23    }
24
25 }
26
```

Screenshots of Running Application:



The screenshot displays the Eclipse IDE interface. The top toolbar contains various icons for file operations and development tools. Below the toolbar, the 'Project Explorer' shows four open files: Logger.java, AsteriskLogger.java, SpacedLogger.java, and App.java. The 'App.java' file is selected and its code is visible in the editor. The code defines a public class 'App' with a main method. Inside the main method, two logger objects are instantiated: 'testAsteriskLogger' of type 'AsteriskLogger' and 'testSpacedLogger' of type 'SpacedLogger'. The 'testAsteriskLogger.log' method is called with the argument 'GoodBye'. Then, 'System.out.println()' is called. Finally, the 'testAsteriskLogger.error' method is called with the argument 'GoodBye!!!'. Below the code editor, the 'Console' tab is active, showing the output of the application. The output consists of four lines, each preceded by a line of asterisks. The first line is '***GoodBye***'. The second line is '***Error: GoodBye!!!***'. The third line is '***G o o d B y e ***'. The fourth line is '***Error: G o o d B y e ! ! ! ***'.

```
1
2 public class App {
3
4     public static void main(String[] args) {
5         AsteriskLogger testAsteriskLogger = new AsteriskLogger();
6         SpacedLogger testSpacedLogger = new SpacedLogger();
7
8         testAsteriskLogger.log("GoodBye");
9
10        System.out.println();
11
12        testAsteriskLogger.error("GoodBye!!!");
13    }
14 }
```

<terminated> App [Java Application] C:\Users\sbotc\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full...

GoodBye

Error: GoodBye!!!

***G o o d B y e ***

***Error: G o o d B y e ! ! ! ***

URL to GitHub Repository:

<https://github.com/DGolf1313/Week5HomeWorkLogger>