PROMINEO TECH

Intro to Java Week 5 Coding Assignment

URL to GitHub Repository: https://github.com/vlanzilo87/Promineo/tree/main/Week5

URL to Public Link of your Video: https://youtu.be/ELY-H8V6Sfo

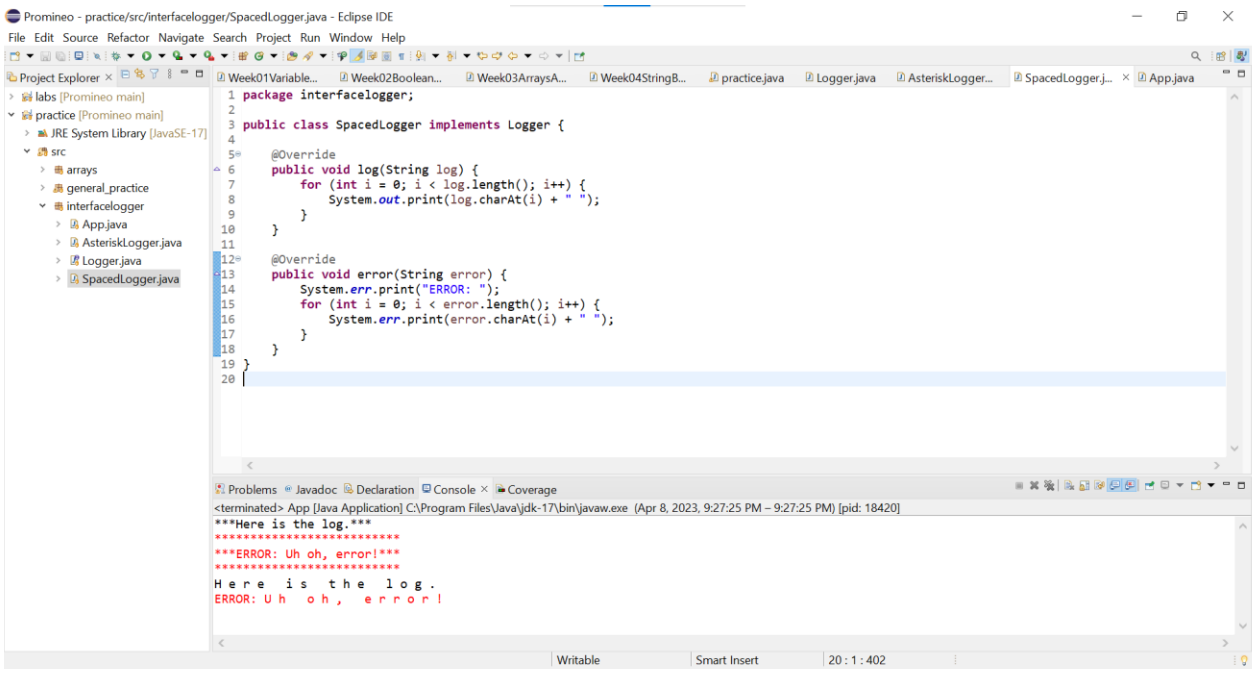
———————————————————————————————————————

Instructions:

1. Follow the **Coding Steps** below to complete this assignment.
    - 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.
    - Create a new repository on GitHub for this week's assignment and push your completed code to this dedicated repo.
    - Create a video showcasing your work:
        - In this video: record and present your project verbally while showing the results of the working project.
        - <u>Easy way to Create a video</u>: Start a meeting in Zoom, share your screen, open Eclipse with the code and your Console window, start recording & record yourself describing and running the program showing the results.
        - Your video should be a maximum of 5 minutes.
        - Upload your video with a public link.
        - <u>Easy way to Create a Public Video Link</u>: Upload your video recording to YouTube with a public link.
2. In addition, please include the following in your Coding Assignment Document:
    - The URL for this week's GitHub repository.
    - The URL of the public link of your video.
3. Save the Coding Assignment Document as a .pdf and do the following:
    - Push the .pdf to the GitHub repo for this week.
    - Upload the .pdf to the LMS in your Coding Assignment Submission.

———————————————————————————————————————

Intro to Java Week 5 Coding Assignment

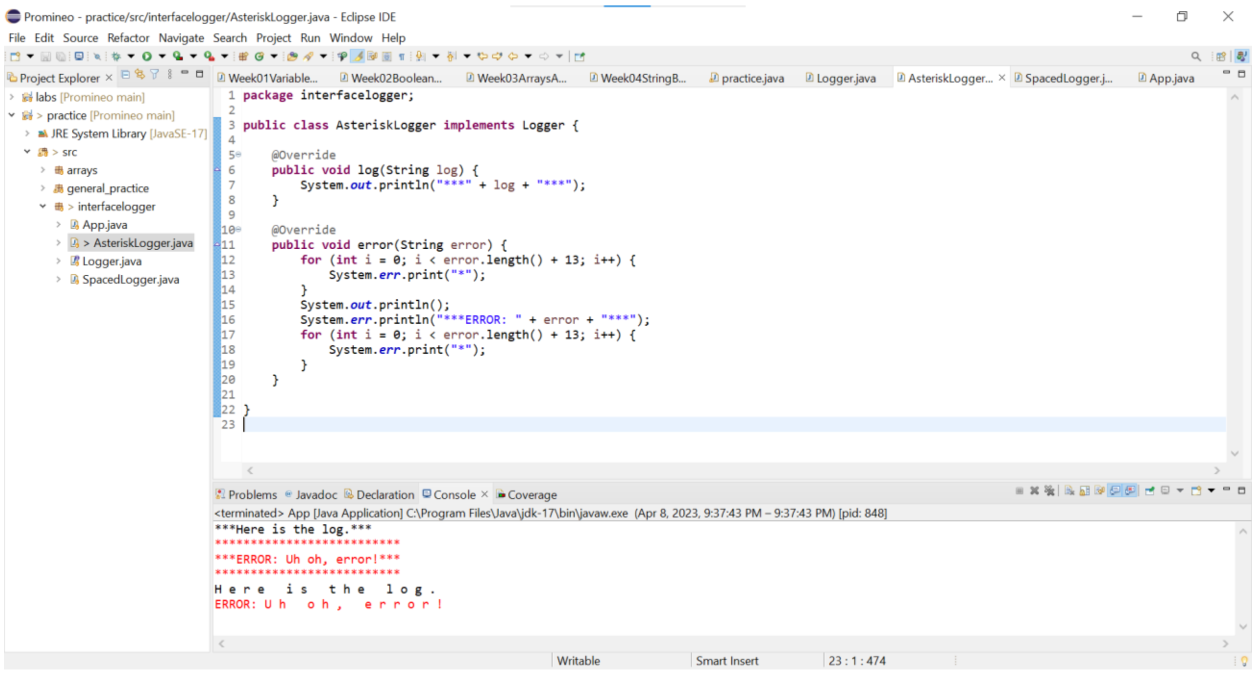


Promineo - practice/src/interfacelogger/SpacedLogger.java - Eclipse IDE

```java
package interfacelogger;

public class SpacedLogger implements Logger {

    @Override
    public void log(String log) {
        for (int i = 0; i < log.length(); i++) {
            System.out.print(log.charAt(i) + " ");
        }
    }

    @Override
    public void error(String error) {
        System.err.print("ERROR: ");
        for (int i = 0; i < error.length(); i++) {
            System.err.print(error.charAt(i) + " ");
        }
    }
}

```

<terminated> App [Java Application] C:\Program Files\Java\jdk-17\bin\javaw.exe (Apr 8, 2023, 9:27:25 PM – 9:27:25 PM) [pid: 18420]

```
***Here is the log.***
**************************
***ERROR: Uh oh, error!***
**************************
H e r e   i s   t h e   l o g . 
ERROR: U h   o h ,   e r r o r ! 
```



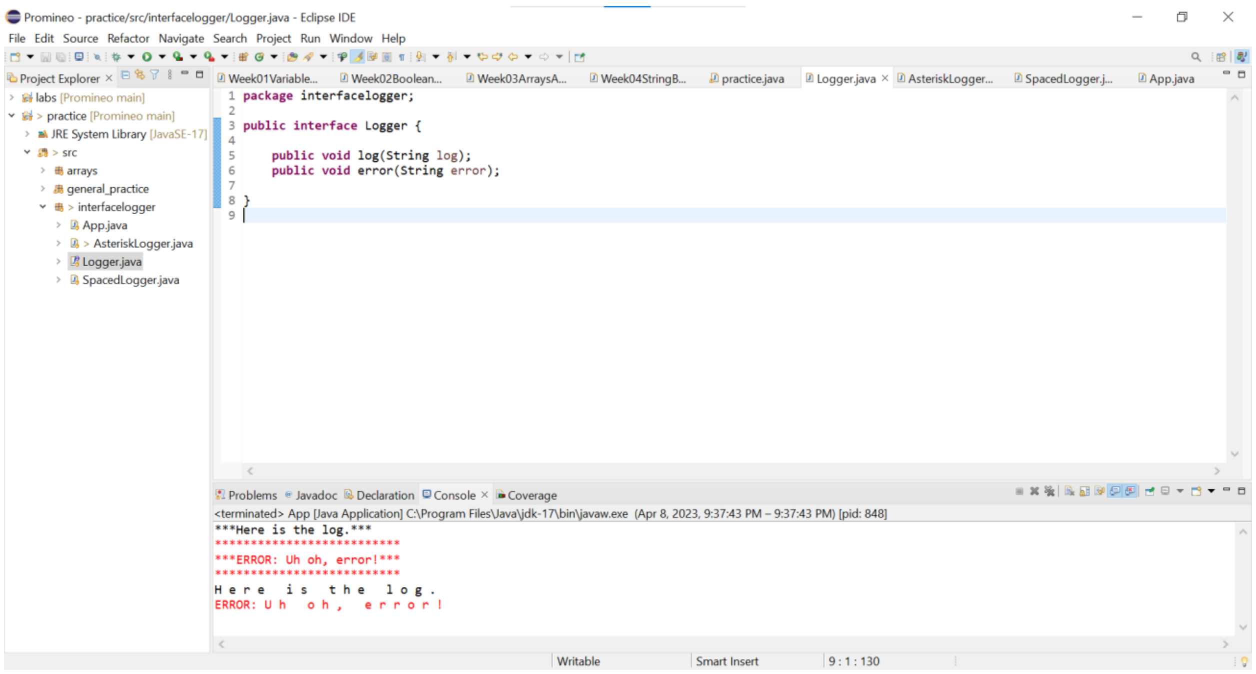
Promineo - practice/src/interfacelogger/AsteriskLogger.java - Eclipse IDE

```java
package interfacelogger;

public class AsteriskLogger implements Logger {

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

    @Override
    public void error(String error) {
        for (int i = 0; i < error.length() + 13; i++) {
            System.err.print("*");
        }
        System.out.println();
        System.err.println("***ERROR: " + error + "***");
        for (int i = 0; i < error.length() + 13; i++) {
            System.err.print("*");
        }
    }
}

```

<terminated> App [Java Application] C:\Program Files\Java\jdk-17\bin\javaw.exe (Apr 8, 2023, 9:37:43 PM – 9:37:43 PM) [pid: 848]

```
***Here is the log.***
**************************
***ERROR: Uh oh, error!***
**************************
H e r e   i s   t h e   l o g . 
ERROR: U h   o h ,   e r r o r ! 
```

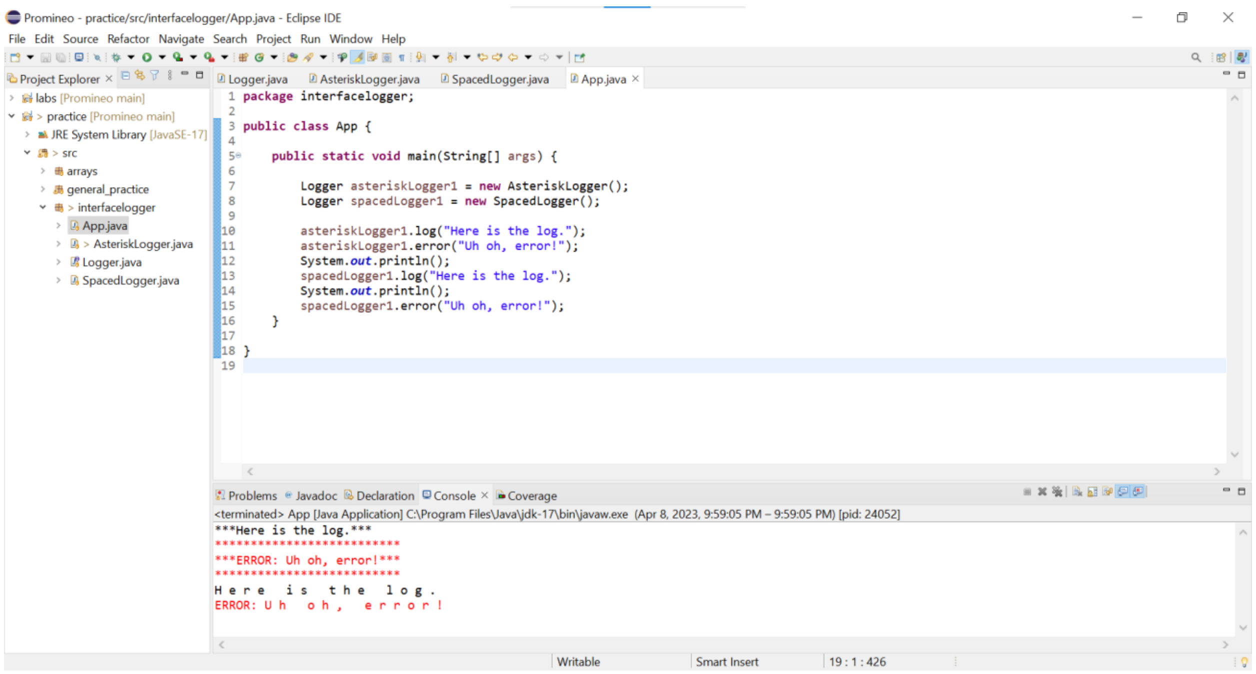
Intro to Java Week 5 Coding Assignment



Promineo - practice/src/interfacelogger/Logger.java - Eclipse IDE

```java
package interfacelogger;

public interface Logger {

    public void log(String log);
    public void error(String error);

}
```

```
<terminated> App [Java Application] C:\Program Files\Java\jdk-17\bin\javaw.exe  (Apr 8, 2023, 9:37:43 PM – 9:37:43 PM) [pid: 848]
***Here is the log.***
**************************
***ERROR: Uh oh, error!***
**************************
H e r e   i s   t h e   l o g .
ERROR: U h   o h ,   e r r o r !
```



Promineo - practice/src/interfacelogger/App.java - Eclipse IDE

```java
package interfacelogger;

public class App {

    public static void main(String[] args) {

        Logger asteriskLogger1 = new AsteriskLogger();
        Logger spacedLogger1 = new SpacedLogger();

        asteriskLogger1.log("Here is the log.");
        asteriskLogger1.error("Uh oh, error!");
        System.out.println();
        spacedLogger1.log("Here is the log.");
        System.out.println();
        spacedLogger1.error("Uh oh, error!");
    }

}
```

```
<terminated> App [Java Application] C:\Program Files\Java\jdk-17\bin\javaw.exe  (Apr 8, 2023, 9:59:05 PM – 9:59:05 PM) [pid: 24052]
***Here is the log.***
**************************
***ERROR: Uh oh, error!***
**************************
H e r e   i s   t h e   l o g .
ERROR: U h   o h ,   e r r o r !
```

PROMINEO TECH

Intro to Java Week 5 Coding Assignment

Coding Steps — Object Oriented Programming:

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.