



Intro to Java Week 5 Coding Assignment

URL to GitHub Repository: <https://github.com/rcooley1/CodingAssignmentW5>

URL to Public Link of your Video:

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.
 - Easy way to Create a video: 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.
 - Easy way to Create a Public Video Link: 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

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.