

As with assignment 1, develop a simple set of Java classes using the Observer pattern to represent the following scenario:

In the land of Middle Earth battle wages between good and bad. On the side of 'good' are Hobbits, Elves, Dwarves, and Men. On the side of 'bad' are the wizard Saruman and the Witch King of Angmar (leader of the Nazgul). The ever watching Eye of Sauron is always on the lookout for the agents of 'good'. Saruman and the Witch King would like to be notified any time the Eye of Sauron spots one of the good guys.

Use the Observer pattern to model the above. Properly utilize Java's Observer interface and Observable class as part of your implementation. Do try this out and see if you can get it to work. NOTE: Depending on the version of the Java you use, you may get a warning that Observer and Observable are deprecated. This is NOT an error and it is ok to proceed with your solution. Use these steps to help you complete this assignment. Follow the Weather Data example from HFDP to aid you in your task.

Use the tester class given below as a starting point for your design (you may add more to the tester as necessary based on your solution to this problem).

Here is some code to get you started:

```
public class TestSauronEye {  
    public static void main(String[] args) {  
        // TODO Auto-generated method stub  
        EyeOfSauron eye = new EyeOfSauron();  
        BadGuy saruman = new BadGuy(eye, "Sauruman");  
        BadGuy witchKing = new BadGuy(eye, "Witch King");  
        GoodGuys gg = new GoodGuys(1,1,2,0);  
        eye.enemiesSpotted(gg);  
        saruman.defeated();  
        gg = new GoodGuys(4,2,2,100);  
        eye.enemiesSpotted(gg);  
    }  
}
```

Include the following in your .zip folder:

Zip folder name cscd212as2\_firstname\_lastname (e.g. cscd212as2\_Nancy\_Ripplinger.zip). Includes the following:

Source files: For Java, the file with main in it should be named TestSauronEye.java. (50 points) Also make sure you can compile and run from the top directory of the unzipped submission. More specifically you should be able to compile and run your code from directory created by unzipping the submission. Send the zipfile to yourself and make sure you can open and run it from Eclipse. We need to become familiar with Eclipse in CSCD212 so run all projects using eclipse.

UML Class diagram (.pdf format required, name file Observer\_UML)

Output capture: include a capture of the output from running your program. Name it Observer\_Output (and include an appropriate extension based on file type -- .txt, .pdf, etc.)

You may use packages or one file it is your choice (flexible) THANK YOU!

Grading: Review rubric for grading criteria