

Web Page Update Notifier  
Due May 7

There are times that one would like to know when a web page has been updated. For example it would be nice to know when a page in the course wiki has changed. We will build a small system to notify people when web pages have been updated. Your program should read a file containing a list of URL of webpages. Each url starts a new line in the file. Each URL is followed by a space then a list of observers for the url. Observers in the list are separated by spaces. We will start with two observers, mail and transcript, but more may be added before the assignment is due. The following are some examples:

```
http://www.eli.sdsu.edu/courses/spring15/cs635/notes/index.html transcript
http://www.eli.sdsu.edu/index.html mail transcript
http://bismarck.sdsu.edu/CS635/recent mail
http://bismarck.sdsu.edu/CS635/8 transcript mail
```

The program should be able to run indefinitely and notify users when one of the pages changes. Additional features are given below.

The headers of a web page have field (Last-Modified) to indicate when they were last modified. However not all web pages provide that information. Another feature one can use is the content length. It is very unlikely that after a page has been modified it will not change size. Some web apps fetch all their content via Javascript so have no content length in the header nor have the last modified date. We can not handle such web pages. In Java the `java.net.URL` class and the `getLastModified()` method in `URLConnection` class to retrieve the last modified date of a web page. That is when the last modified date of the web page changes or is more recent than a given time. There are similar methods for content length. Below is an example of using the `URL` and `URLConnection` classes.

```
URL address = new URL("http://www.eli.sdsu.edu/");
URLConnection connect = address.openConnection();
long time = connect.getLastModified();
Date modifiedDate = new Date(time);
System.out.println(modifiedDate.toString());
```

When a web page has been modified broadcast the change to observers. To start with we will use two different type of observers. The first type will print a notification to the standard output. The second type will send an email message. You can download JavaMail at: <https://java.net/projects/javamail/pages/Home>. Oracle has a JavaMail FAQ at: <http://www.oracle.com/technetwork/java/javamail/faq/index.html>. There are numerous tutorials on using JavaMail on the web.

One problem is that web pages don't change each time you wish to test your program. So we will use factory method and mock objects. Create a factory method that returns a URL object (or a URLConnection object). Create a subclass that returns a mock object for the URL object. The mock URL object will return a mock URLConnection object. The mock URLConnection object returns the proper value for the getLastModified() method. See <http://code.google.com/p/mockito/> for more information on mock objects.

For the program to be useful it needs to run continuously. This increases the chances that the program or OS may crash while the program is running. Use the memento pattern to save the state of your program to a file. When the program starts up it can recover its last state from the file.

### **Grading**

	Percent of Grade
Working Code	15%
Unit Tests	15%
Quality of Code	20%
Proper implementation of Patterns	50%

### **What to Turn in**

Turn in hard copy of your code.

### **Late Policy**

An assignment turned in 1-7 days late, will lose 5% of the total value of the assignment per day late. The eighth day late the penalty will be 40% of the assignment, the ninth day late the penalty will be 60%, after the ninth day late the penalty will be 90%. Once a solution to an assignment has been posted or discussed in class, the assignment will no longer be accepted. Late penalties are always rounded up to the next integer value.