Assignment 1

I have created "skeletons" of a simple web server and a simple web client (not a full web browser, but when completed it will display simple HTML pages). Both skeletons are written in Java SE 8. The skeletons compile, but they do not function as we would like. In particular, both files contain comments that end with the line

```
// >>> YOUR CODE TO ACCOMPLISH THIS GOES NEXT <<<
```

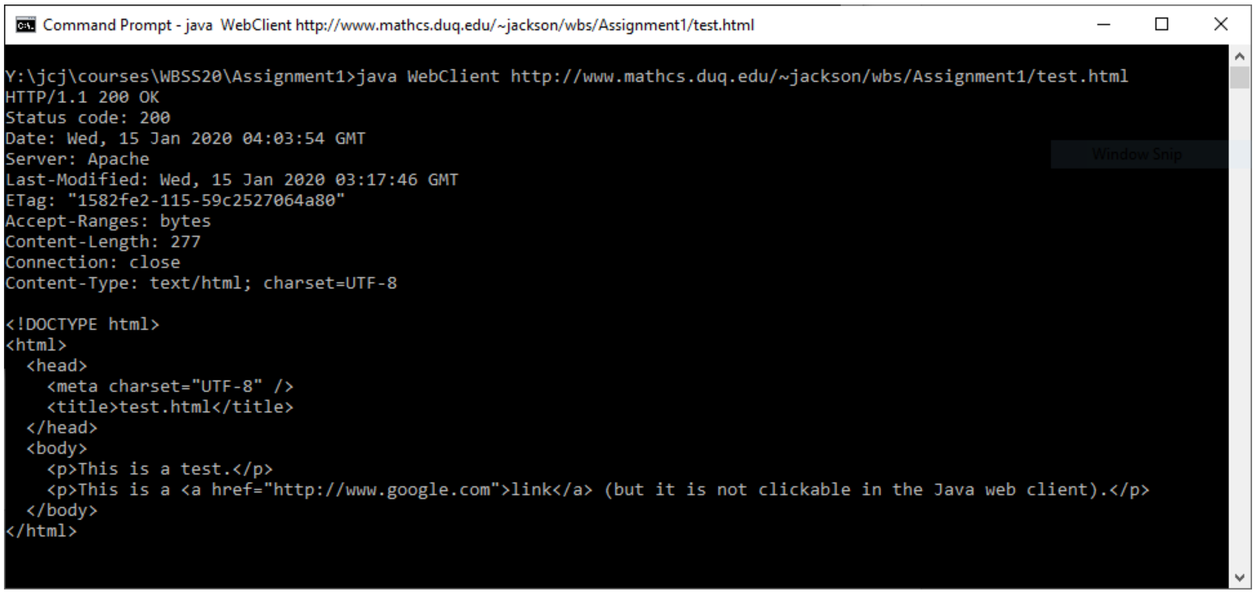
There are two such comments in WebClientSkeleton.java and three in WebServerSkeleton.java. Your assignment is to use these skeleton files to write WebClient.java and WebServer.java, a working web client and server. You'll do this by downloading the provided skeleton code, changing the class names in each file (by removing "Skeleton") and saving the files with the appropriate names, and then filling in code (that runs with Java SE 8, the version of Java running by default on our lab machines) beneath these five comments in order to implement the functionality described by the comments. You should only place code in the five places indicated. This includes not adding any additional import statements. You should also not remove any of the code I have written. Turn in your modified programs, named WebClient.java and WebServer.java, at Blackboard.

Grading: 3 points for each section of code correctly implemented. I might give partial credit for a section that is not correct but reasonably close, but don't count on it. Definitely no credit for a section if you modify my code related to the section, no credit if you use packages other than those imported by the skeletons, and no credit for a program if it does not compile. I may also take off points for not following directions, such as not submitting code at Blackboard, not using the specified class names, etc.

Hints/Helps

- I hope that you'll begin by reading through both skeletons. You'll find many comments that should help you to understand what the skeletons are doing and what functionality you need to add.
- You're encouraged to use the Java SE 8 API documentation found at https://docs.oracle.com/javase/8/docs/api/index.html. This is not considered "help" for Gilligan's Island Rule purposes. The skeletons use many API classes, and you'll probably want to read about the methods of at least some of them to better understand what the code is doing and/or to help you to create the required code.
- I have supplied simple test HTML documents at the following URLs:
  http://www.mathcs.duq.edu/~jackson/wbs/Assignment1/test.html
  http://www.mathcs.duq.edu/~jackson/wbs/Assignment1/test2.html
  I recommend that you test your client only with these files (or similarly simple files); it will not display web pages that use anything more than very basic HTML.
- You can test your client from a command line by entering text such as
  ```
  java WebClient http://www.mathcs.duq.edu/~jackson/wbs/Asssignment1/test.html
  ```
  That is, follow the program name with a space and then the URL of a page to visit. If your client

is working correctly, you should see something very similar to the following in the console:

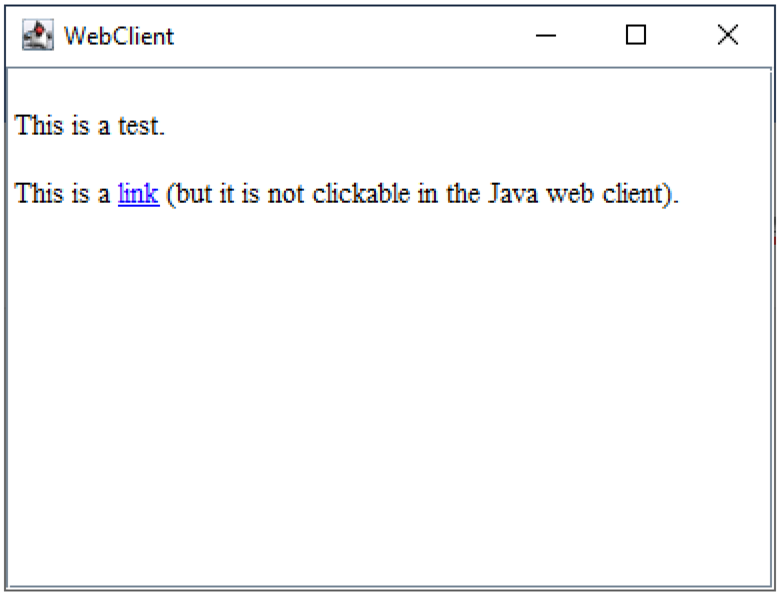


Command Prompt - java WebClient http://www.mathcs.duq.edu/~jackson/wbs/Assignment1/test.html

```
Y:\jcj\courses\WBSS20\Assignment1>java WebClient http://www.mathcs.duq.edu/~jackson/wbs/Assignment1/test.html
HTTP/1.1 200 OK
Status code: 200
Date: Wed, 15 Jan 2020 04:03:54 GMT
Server: Apache
Last-Modified: Wed, 15 Jan 2020 03:17:46 GMT
ETag: "1582fe2-115-59c2527064a80"
Accept-Ranges: bytes
Content-Length: 277
Connection: close
Content-Type: text/html; charset=UTF-8

<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8" />
    <title>test.html</title>
  </head>
  <body>
    <p>This is a test.</p>
    <p>This is a <a href="http://www.google.com">link</a> (but it is not clickable in the Java web client).</p>
  </body>
</html>
```

and a window should appear that looks essentially identical to the following:



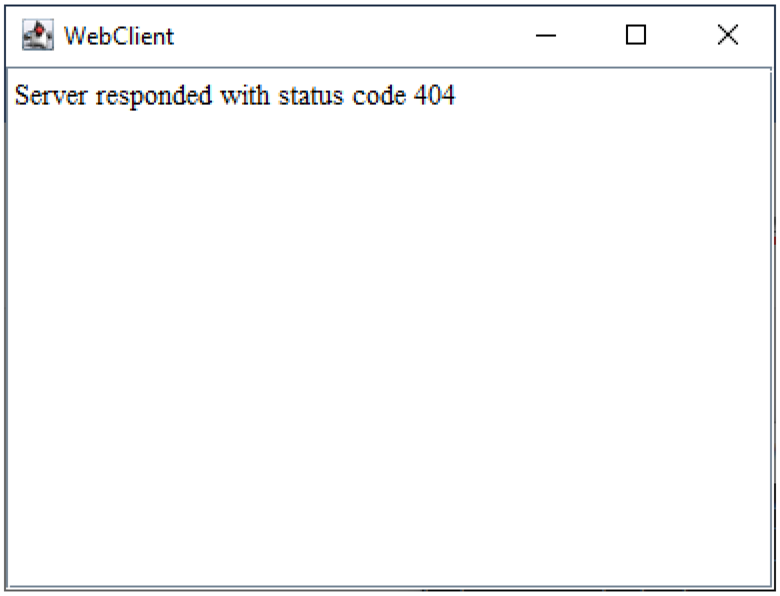
WebClient

This is a test.

This is a link (but it is not clickable in the Java web client).

- Once you start the web client program, it will continue to run (you will not see a prompt in the console) until you close the Java pop-up window.

- You should also test your client on a nonexistent file (such as using test1.html in place of test.html). You should then see a window similar to the following:



WebClient

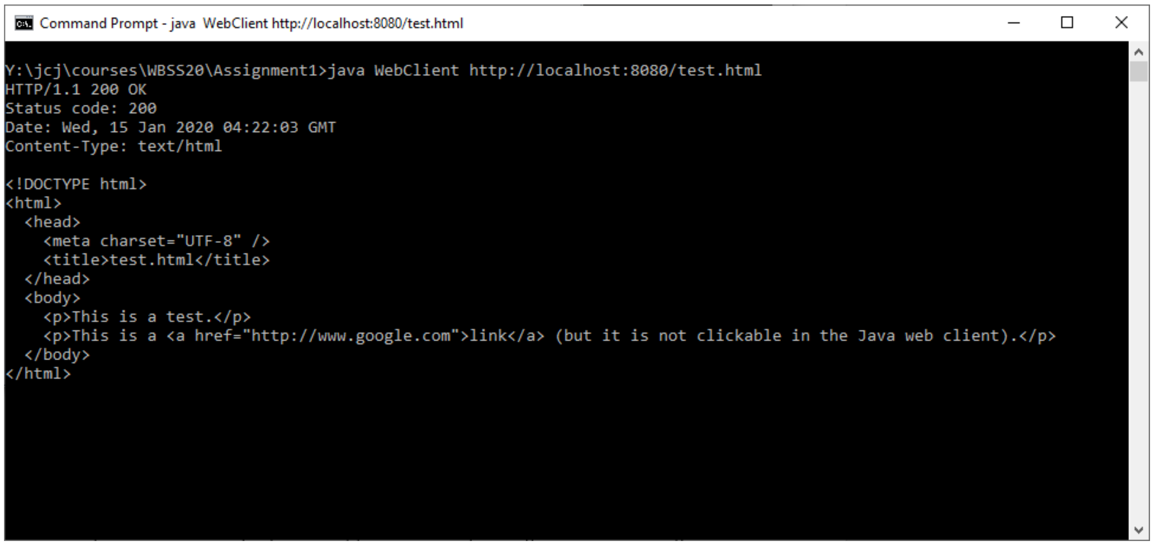
```
Server responded with status code 404
```

- Note that the server will "listen" for connections to port 8080 rather than to the standard HTTP port 80. This is because low-numbered ports are special and typically require that a program have administrative privileges in order to use them, so we'll use a high-numbered port.
- There are several ways to test your server, once you think that you have completed the code for it. First, download test.html from Blackboard and place the file in the same folder as your WebServer.class file. Then, start your server; to do this from the command line, you enter

  ```
  java WebServer
  ```

  You should see a message stating that the server is starting on port 8080. Then you have several options:
  - Browse, in Chrome, Firefox, or any other standard browser, to the URL `http://localhost:8080/test.html`
    If your server is working, you will see a page much like the one shown above displayed in your browser. ("localhost" is a special hostname that represents the machine on which the browser is running.)
  - If your web client code seems to be working, you can run it as follows:
    `java WebClient http://localhost:8080/test.html`
    This should not only pop up the "This is a test" page shown earlier, but it should also

show output in the console similar to the following:


Command Prompt - java WebClient http://localhost:8080/test.html

```
Y:\jcj\courses\WBSS20\Assignment1>java WebClient http://localhost:8080/test.html
HTTP/1.1 200 OK
Status code: 200
Date: Wed, 15 Jan 2020 04:22:03 GMT
Content-Type: text/html

<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8" />
    <title>test.html</title>
  </head>
  <body>
    <p>This is a test.</p>
    <p>This is a <a href="http://www.google.com">link</a> (but it is not clickable in the Java web client).</p>
  </body>
</html>
```

If you do not see the status line, headers, blank line, and body as shown (except for the value of the date), then something is wrong with your server, client, or both.

- Finally, you can telnet to your server. If you have telnet enabled on your machine (on Windows 10, you need to enable it; Google can tell you how), then you can enter something like the following in a command window/console:

```
telnet localhost 8080
GET /test.html HTTP/1.1
```

Press enter twice after the GET line (and note that you might not see the GET line while you are typing if you are using the Windows telnet client; just type carefully!). You should see a response very similar to what is shown in the previous console window. If not, something is wrong with your server.

- If you have one server running on port 8080, you cannot run a second server on that port. So, if you see a message saying something about a bind exception or address already in use, you probably need to kill a running server. You can do this by typing Ctrl-C in the console window where the server is running. Once this is done, you can start a new copy of your server code.
- I had some trouble testing my server with Chrome. Specifically, after Chrome visited my server, if it received a 404 (file not found) response, it seemed to after 30 seconds or so open another connection but not actually send any data over the connection. This behavior caused my server to crash. Firefox didn't give me the same problems. Bottom line: If your server behaves like mine did and crashes with Chrome, it will not count against you.