Project 3

In this assignment, you will develop a simple Web client/server in Java. You can assume the server is capable of processing only one request at a time.

Specifically, your Web server will

1) create a TCP connection socket when contacted by a client (that plays a browser role)
2) receive HTTP GET requests over the connection
3) parse the request to determine the specific file being requested
4) get the requested file from the server's file system
5) create an HTTP response message consisting of the requested file preceded by header lines
6) send the response over the TCP connection to the requesting client

If a browser requests a file that is not present in your server, your server should return a "404 Not Found" error message.

Your Web client will

1) create a connection socket given the server info (IP and Port 80)
2) send an HTTP GET request asking for a file specified by a user input
3) receive the HTTP response message from the server
    a) If it is an error message, show the error
    b) If it is a requested file, it saves the file into a local directory

You can use a simplified HTTP request message format. A request message may look like:

```
GET http://192.168.0.2/files/hello.txt
```

Similarly, a response message may look like:

```
HTTP/1.1 200 OK
[content body]
```

Reference

A simple example of TCP socket programming

Course Canvas: Files > codes > tcp socket example

HTTP GET message format
https://book.systemsapproach.org/applications/traditional.html#request-messages

Output stream
https://docs.oracle.com/en/java/javase/11/docs/api/java.base/java/io/OutputStream.html

- For the actual implementation, check “Direct Known Subclasses”
- Check the role of .flush() and .write()
- .write() only takes byte arrays

Input stream
https://docs.oracle.com/en/java/javase/11/docs/api/java.base/java/io/InputStream.html

Note

- You should not use the Java HttpClient or HttpServer classes. This project is intended to recreate a simplified version of these classes.
- You can prepare image/text/pdf files on your server machine. Test if your requests to the different files are served correctly.

Submission

- For both client and server, please use the standard terminal output to log the internal states like:
  - Socket created with IP/Port
  - A message sent
  - A message received
  - What is the parsed HTTP request
  - What is the response message content
- You do not need to write a report. Submit a PDF file including the following screen captures with appropriate headlines. (These captures should show the log of the internal states, too.)
  - Show screen captures (both server and client) that show the client can download and save
    - Image file (.png or .jpg),
    - A simple text file,
    - Our course slide PDF file
  - Show screen captures (both server and client) that show the client receives an error message when requesting an invalid file name.

- Along with the pdf file including the terminal outputs, please submit your source codes.
    - Please comment on the codes appropriately so the grader can understand your codes easily.