Homework 2
TCP Socket Programming/Webserver

PROGRAM SPECIFICATIONS

In this lab, you will learn the basics of socket programming for TCP connections in Python: how to create a socket, bind it to a specific address and port, as well as send and receive a HTTP packet. You will also learn some basics of HTTP header format.

You will develop a web server that handles one HTTP request at a time. Your web server should accept and parse the HTTP request, get the requested file from the server's file system, create an HTTP response message consisting of the requested file preceded by header lines, and then send the response directly to the client. If the requested file is not present in the server, the server should send an HTTP "404 Not Found" message back to the client.

CODE

Below you will find the skeleton code for the Web server. You are to complete the skeleton code. The places where you need to fill in code are marked with #Fill in start and #Fill in end. Each place may require one or more lines of code.

RUNNING THE SERVER

Included in this lab is a simple HelloWorld.html file. The file will need to go into the same directory as the webserver.

To run the webserver

1) Open a command line terminal
2) Change directories to where Webserver.py is stored
3) Run the server with python Webserver.py

NOTE: The IP address of the webserver should be 127.0.0.1 and the port should be 6789.

RUNNING THE CLIENT

In this case the client is your favorite web browser.

To run the client

1) Open your favorite web browser
2) Enter the URL http://127.0.0.1:6789/hello.html

NOTE: I could only run the code in chrome. Firefox always threw an error regardless of what I was trying.

NOTE: If you omit the port of 6789, the browser will assume port 80 and you should get **This site can't be reached message**

RUNNING THE CLIENT AGAIN

In this case the client is your favorite web browser.

To run the client

1) Open your favorite web browser
2) Enter the URL http://127.0.0.1:6789/helloBAD.html

You should get a **404 Not Found** message.

SKELETON PYTHON CODE

Remember indentation is really important in Python. Please note how this code is indented. I thought about it and it is best for your to type your own program instead of trying to edit my Webserver.py file.

```
#import socket module
from socket import *

import sys # In order to terminate the program

serverSocket = socket(AF_INET, SOCK_STREAM)

#Prepare a sever socket to bind to an IP and listen for one connection
#Fill in start

#Fill in end

while True:
        #Establish the connection
        print('The server is ready to receive')

        # Set up a new connection from the client
        connectionSocket, addr = #Fill in start #Fill in end

        # If an exception occurs during the execution of try clause
        # the rest of the clause is skipped
        # If the exception type matches the word after except
        # the except clause is executed
        try:
                # Receives the request message from the client (bytes and the message will need to be decoded)
                message = #Fill in start #Fill in end

                # Extract the path of the requested object from the message
                # The path is the second part of HTTP header, identified by [1]
                filename = message.split()[1]

                # Because the extracted path of the HTTP request includes
                # a character '\', we read the path from the second character
                f = open(filename[1:])

                # Store the entire contents read of the requested file in a temporary buffer
                outputdata = #Fill in start #Fill in end

                #Send one HTTP header line into socket
                #Fill in start

                #Fill in end

                #Send the content of the requested file to the client
                for i in range(0, len(outputdata)):
                        connectionSocket.send(outputdata[i].encode())
                connectionSocket.send("\r\n".encode())

                # Close the client connection socket
                connectionSocket.close()

        except IOError:
                # Send HTTP response message for file not found (http type and status code) – you will need to encode this
                # Send the HTTP message too you will need to encode this
                #Fill in start

                #Fill in end

                #Close client socket
                #Fill in start

                #Fill in end

serverSocket.close()
sys.exit()#Terminate the program after sending the corresponding data
```

TO COMPLETE

1) Complete the Python code that creates a functioning webserver. You can copy and paste, or you can type it. My recommendation is to type it is easier then copy and pasting and trying to fix code.
2) Run the webserver
3) Execute the client with valid IP, port and hello.html – capture the output of the webserver and the browser
4) Execute the client with valid IP, port and helloBAD.html – Capture the output of the webserver and the browser

TO TURN IN

A zip file that contains

1) A PDF of your Python code and your runs
2) Your Webserver.py file
3) The hello.html file

Name your zip with your last name first letter of your first name hw2.zip (Example: steinershw2.zip)