

# CSC361 Tutorial 3

Assignment Project Exam Help

<https://tutorcs.com>

WeChat: cstutorcs

- Marking Breakdown
- Test Scenarios
- P1 Demo
- Design Check

# Marking Breakdown

- 200 ok 1 pt
- Persistent connection 1pt
- Correct return large file 1pt
- 404 not found 1pt
- 400 bad request 2 pts
- Multiple clients 1 pt
- Timeout 1 pt
- Back-to-back request 1pt
- Server output format 1pt

Assignment Project Exam Help

<https://tutorcs.com>

WeChat: cstutorcs

# Test Scenario 1

- 200 ok test

- Small file request
- Test whether 200 ok responded correctly 1pt
- Test the format of server output 1pt
- Large file request, if return content correctly 1pt

Assignment Project Exam Help

<https://tutorcs.com>

WeChat: cstutorcs

## Test Scenario 2

- 404 test

- Non-existing file request
- Test whether 404 responded correctly [p]

Assignment Project Exam Help

<https://tutorcs.com>

WeChat: cstutorcs

## Test Scenario 3

- 400 bad request test
  - Test whether 400 bad request responded correctly. 2pt

Assignment Project Exam Help

<https://tutorcs.com>

WeChat: cstutorcs

# Test Scenario 4

- Multi-client test

- On h1 send a persistent connection, and on r send another 1pt

Assignment Project Exam Help

<https://tutorcs.com>

WeChat: cstutorcs

# Test Scenario 5

- Persistent and timeout test
  - On h1 send a persistent connection, then send another request without connection header 1pt
  - On h1 send a persistent connection, then wait for timeout 1pt

Assignment Project Exam Help

<https://tutorcs.com>

WeChat: cstutorcs

# Test Scenario 6

- Back-to-back test
  - On h1 send requests from a text file 1pt

Assignment Project Exam Help

<https://tutorcs.com>

WeChat: cstutorcs



- To test your server whether this feature is enabled, you can create a text file (call test.txt) as follows:

```
GET /notfound HTTP/1.0  
Connection:keep-alive
```

```
GET /hello.html HTTP/1.0  
Connection:keep-alive
```

```
get /hello.html http/1.0  
Connection:keep-alive
```

```
GET /notfound HTTP/1.0  
Connection:keep-alive
```

- nc ip\_address port < test.txt

Assignment Project Exam Help

<https://tutorcs.com>

WeChat: cstutorcs

# Demo 1: Line by line Requests

use nc connect to sws, and input requests and headers line by line

Assignment Project Exam Help

<https://tutorcs.com>  
root@h1:/home/jovyan/p1\_demo# nc 10.10.1.100 8080  
GET /small.html HTTP/1.0  
[

WeChat: cstutorcs

## Demo 2: Back-to-back requests

Send multiple requests at the same time:

Assignment Project Exam Help

```
root@h1:/home/jovyan/pl_demo# nc 10.10.1.100 8080 < multiple_requests.txt
HTTP/1.0 404 Not Found
Connection: keep-alive

HTTP/1.0 200 OK
Connection: close
```

<https://tutorcs.com>

WeChat: cstutorcs

# Common Problems

1. Server output log format
2. Fail to respond to bad request immediately (do not need to wait for the new line)
3. Cannot support line by line and back-to-back requests simultaneously
4. HTTP Headers
5. Close the connection when
  - a. No keep-alive headers in the request
  - b. Bad requests
  - c. timeout
6. Extra new line returned after the end of the file content, which is not correct

Assignment Project Exam Help

<https://tutorcs.com>

WeChat: cstutorcs

```
server_address = ('localhost', 80)
server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
server.setblocking(0)
server.bind(server_address)
```

Assignment Project Exam Help

<https://tutorcs.com>

You should not hardcode the ip address and port number

WeChat: cstutorcs

if not is\_match:

```
response_messages[s] += "HTTP/1.0 400 Bad Request"
```

else:

**Assignment Project Exam Help**

```
response_messages[s] += "HTTP/1.0 200 OK"
```

**<https://tutorcs.com>**

**WeChat: cstutorcs**

Be sure to consider the case of 404 not found

Do not forget to add “\r\n” for the response line

```
# keep-alive is false, close the connection
if (connection_alive == None):
    print("closing connection...")
    close_socket(sock, inputs, outputs, message_queues)
```

Assignment Project Exam Help

<https://tutorcs.com>

WeChat: cstutorcs

Remember to remove any unnecessary prints before submission.