

## CS352 Project 1

**1. Team details: Clearly state the names and netids of your team members (there are 2 of you).**

Jing Jia (jj740)

Dong Shu (ds1657)

**2. Collaboration: Who did you collaborate with on this project? What resources and references did you consult? Please also specify on what aspect of the project you collaborated or consulted.**

Jing Jia and Dong Shu collaborate on this project.

We attended recitation and asked the TA regarding some problems we encountered.

We also asked the professor regarding the data type issue after class.

We also used Real Python as a reference for the socket functions, such as send() and recv().

“Socket Programming in Python (Guide).” *Realpython.com*, Real Python, 21 Feb. 2022, [realpython.com/python-sockets/#socket-api-overview](https://realpython.com/python-sockets/#socket-api-overview).

**3. What did you observe after running step 2 above? Can you explain why you see what you see?**

We commented out the two sleep statements.

```
if __name__ == "__main__":
    t1 = threading.Thread(name='server', target=server)
    t1.start()

    #time.sleep(random.random() * 5)
    t2 = threading.Thread(name='client', target=client)
    t2.start()

    #time.sleep(5)
    print("Done.")
```

### First Run

```
● jj740@frost:~/Desktop/Internet Tech$ python proj.py
[S]: Server socket created
[S]: Server host name is frost.cs.rutgers.edu
[C]: Client socket created
Done.
[S]: Server IP address is 128.6.13.150
[S]: Got a connection request from a client at ('128.6.13.150', 35468)
[C]: Data received from server: Welcome to CS 352!
```

### Second Run

```
● jj740@frost:~/Desktop/Internet Tech$ python proj.py
[S]: Server socket created
[C]: Client socket created
Done.
Exception in thread server:
Traceback (most recent call last):
  File "/usr/lib/python2.7/threading.py", line 801, in __bootstrap_inner
    self.run()
  File "/usr/lib/python2.7/threading.py", line 754, in run
    self._target(*self._args, **self._kwargs)
  File "proj.py", line 16, in server
    ss.bind(server_binding)
  File "/usr/lib/python2.7/socket.py", line 228, in meth
    return getattr(self._sock,name)(*args)
error: [Errno 98] Address already in use

Exception in thread client:
Traceback (most recent call last):
  File "/usr/lib/python2.7/threading.py", line 801, in __bootstrap_inner
    self.run()
  File "/usr/lib/python2.7/threading.py", line 754, in run
    self._target(*self._args, **self._kwargs)
  File "proj.py", line 47, in client
    cs.connect(server_binding)
  File "/usr/lib/python2.7/socket.py", line 228, in meth
    return getattr(self._sock,name)(*args)
error: [Errno 111] Connection refused
```

In the first run, the “Done” is printed prior to program termination. The purpose of the first `time.sleep()` is to make sure the client connects online after the server has been returned. The purpose of the second `time.sleep()` is to make sure “Done.” is printed after the client has been returned. However, after we return the first `time.sleep()` has been removed, the client will connect online before the server has returned. In this case, the server and client will be online at the

approximate same time. Moreover, before the client has been returned, the “Done.” will be printed. And that is what we think happens during the first run.

In the second run, the error messages “Address in use” and “Connection refused” are printed. The error is due to the source not released on time in the second run. The Python program is the user level function and after running programming, the process will move to the kernel layer to allocate the resource for the program. In the first run, a port is requested and then occupied for transferring data. When the program is terminated, a signal will be sent to the transport layer to release the resource, however, this port will not be released right after the “Done” is printed to ensure all the data is transferred. Therefore, in the second run, the port is still not released at the moment, which will return the message “Address in use” and “Connection refused”.

#### 4. Is there any portion of your code that does not work as required in the description above? Please explain.

We believe our code works as required in the description as shown below.

```
● jj740@frost:~/Desktop/Internet Tech$ python client.py
[C]: Client socket created
[C]: Data received from server: Welcome to CS 352!
[C]: Data sent to server: HELLO
Eva, Can I Stab Bats In A Cave
Patriots win all the time!!
Madam In Eden, I'm Adam
a socket needs IP address and port
[C]: Data received from server: OLLEH
Done.
-

● jj740@frost:~/Desktop/Internet Tech$ python server.py
[S]: Server socket created
[S]: Server host name is frost.cs.rutgers.edu
[S]: Server IP address is 128.6.13.150
[S]: Got a connection request from a client at ('128.6.13.150', 35428)
[S]: Data received from client: OLLEH
[S]: Data sent to client: OLLEH
-
```

≡ in-proj.txt

```
1 Eva, Can I Stab Bats In A Cave
2 Patriots win all the time!!
3 Madam In Eden, I'm Adam
4 a socket needs IP address and port
5
```

≡ out-proj.txt

```
1 evaC A nI staB bats I naC ,avE
2 !!emit eht lla niw stoirtaP
3 madA m'I ,nedE nI madaM
4 trop dna sserdda PI sdeen tekcos a
5
```

#### 5. Did you encounter any difficulties? If so, explain.

We had problem running the code “csockid.recv(200)”. On Jing Jia’s computer, the return type of this code is string. However on Dong Shu’s computer, the return type is bytes, which affects the

following code where we should send the data to the client. Since our return types are different, we have no idea if we should add the encode function to the data. After asking the professor, we decide to add `str()` function to the data, so that it will guarantee to have a string type.

**6. What did you learn from working on this project? Add any interesting observations not otherwise covered in the questions above. Be specific and technical in your response.**

A interesting observation is that when we sent the “in-proj.txt” data from client to server, sometimes the client.py will combined two lines of data together and sent it to the server. Thus, when we compiled the code, one of the lines will combine to the previous line where it is supposed to appear independently.

Our solution is that we add “`time.sleep(0.5)` ” after each “`cs.send()`”. The client will pause for 0.5 sec before it sends each line. In this case, we do not need to worry about the client combining our data accidentally.