

# Computer Network Lab

## CSE-325

---

### Assignment - 3

Submitted by -

Gyanendra Shukla

CSE 1

191112040

#### To Write a Socket Program to implement ECHO

An echo server is a program that listens for a message from a client and then sends the same message back to the client.

I've implemented the server in python. The server listens on port 12345 for any message and returns the message with current timestamp prepended.

#### Server

```
1  from datetime import datetime
2  import socket
3
4  class Server:
5      def __init__(self, port) -> None:
6          self.port = port
7          self.sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
8          self.sock.bind(('', port))
9          self.sock.listen()
10         print('Server started on port {}'.format(port))
11
12     def run(self) -> None:
13         while True:
14             conn, addr = self.sock.accept()
15             print('Got connection from', addr)
16             with conn:
17                 while True:
18                     data = conn.recv(1024)
19                     if not data:
20                         break
21                     else:
22                         print(f'Got: {data.decode()}')
23                         cur_time = datetime.now()
24                         x = f'{cur_time} : {data.decode()}'
25                         conn.sendall(x.encode())
26
27
28     def __del__(self) -> None:
29         self.sock.close()
30         print('Server closed')
31
```

```
32
33 if __name__ == '__main__':
34     server = Server(12345)
35     server.run()
36
```

## Client

```
1  import socket
2
3  class Client:
4      def __init__(self, host, port) -> None:
5          self.host = host
6          self.port = port
7          self.sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
8          self.sock.connect((host, port))
9          print('Connected to {}:{}'.format(host, port))
10
11     def run(self) -> None:
12         while True:
13             data = input('> ')
14             if not data:
15                 break
16             self.sock.sendall(data.encode())
17             data = self.sock.recv(1024)
18             print(data.decode())
19
20     def __del__(self) -> None:
21         self.sock.close()
22         print('Connection closed')
23
24
25 if __name__ == '__main__':
26     client = Client('127.0.0.1', 12345)
27     client.run()
```

## Output

```

Server started on port 12345
Got connection from ('127.0.0.1', 30369)
Got: Hello, server
Got: I'm sending a message!
Got: My name is Gyanendra!!!!!!!!!!!!!!
Got: 191112040
Got connection from ('127.0.0.1', 30385)
Got: m
Got: e
Got: s
Got: s
Got: a
Got: g
Got: e
Got:
Got: t
Got: h
Got: r
Got: o
Got: u
Got: g
Got: h
Got:
Got: t
Got: e
Got: l
Got: e
Got: n
Got: t
Got:

```

Fig: Messages received by the server

```

In D:\Books\sem 6\networks\labs\lab3 python .\client.py
Connected to 127.0.0.1:12345
> Hello, server
2022-01-30 21:02:37.689554 : Hello, server
> I'm sending a message!
2022-01-30 21:02:44.336316 : I'm sending a message!
> My name is Gyanendra!!!!!!!!!!!!!!
2022-01-30 21:03:06.564357 : My name is Gyanendra!!!!!!!!!!!!!!
> 191112040
2022-01-30 21:03:11.212423 : 191112040

```

Fig: Messages sent by the client and then messages timestamped by the server returned to the client

```

Telnet localhost
2022-01-30 21:03:52.417424 : me2022-01-30 21:03:53.436225 : es2022-01-30 21:03:54.300817 : ss2022-01-30 21:03:54.548335
: sa2022-01-30 21:03:55.094188 : ag2022-01-30 21:03:55.402846 : ge2022-01-30 21:03:55.482547 : e 2022-01-30 21:03:56.49
2405 : t2022-01-30 21:03:57.214852 : th2022-01-30 21:03:57.423168 : hr2022-01-30 21:03:57.605966 : ro2022-01-30 21:03:5
7.694965 : ou2022-01-30 21:03:57.773127 : ug2022-01-30 21:03:57.978593 : gh2022-01-30 21:03:58.108031 : h 2022-01-30 21:
03:58.793041 : t2022-01-30 21:03:59.413735 : te2022-01-30 21:03:59.542941 : el2022-01-30 21:03:59.671223 : le2022-01-30
21:03:59.882320 : en2022-01-30 21:04:00.063999 : nt2022-01-30 21:04:00.214223 : t
2022-01-30 21:04:01.834790 :

```

Fig: Connecting on server through telnet

