Computer Systems Administration

Semester 2 Python Assignment 2020

Student ID: A00268440
Course/Year: Computer Engineering 3

Assignment Title: Basic Encryption with Python

Brief Description

The purpose of this client-server application is to provide security of chat messages and save private information as an encrypted message.

Contents

Detailed Description

This program aims to secure the messages between client and server.
On the server side, the process works by passing input messages to a defined hash function. And the hash function returns a hash string that keeps the encrypted information.

On the client side, both the encrypted message and the original message entered by the user are displayed on the screen. That means the user is allowed to see both messages.

Encryption secures online information and protects governments from countless daily attacks and is a vital component of the national defences.
What I tried to do is a basic form of a securing system using encryption.

Python Features

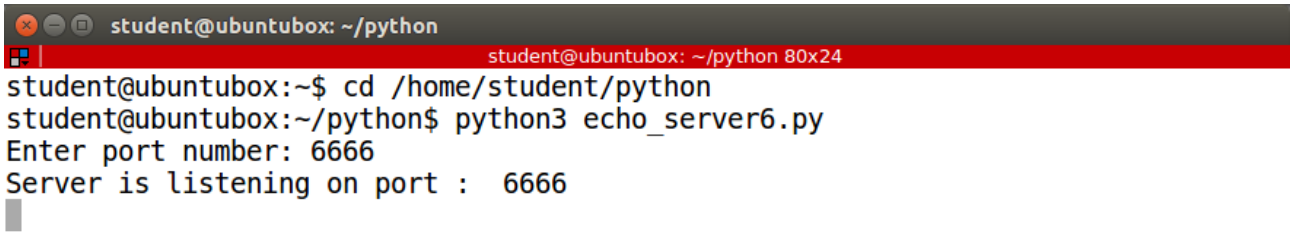
Features	**Yes / No**
The features from the 5 basic Echo exercises Command-line arguments (for server address/port number) Exception Handling (for client connecting/server binding) Transport Layer Security	**Yes**
Client GUI (EasyGUI only)	**Yes**
Functions and Threading Specify Function names and purpose Data Structures (Lists, Dictionaries/Shelves) Specify Data Structures names and purpose	**Function Yes** **Threading No** **List Yes** **Dictionary Yes** **Shelves No**
Additional command-line arguments Specify the purpose of each:	**No**
Additional exception handling Specify: Additional features in the client Specify: Additional features in the server Specify:	**1- ValueError:** **When the user enters an inappropriate type of input** **2- KeyboardInterrupt:** **When the user hits an interrupt key** **No** **Yes** **Hashing function to converting the input message into a encrypted message**

User Manual

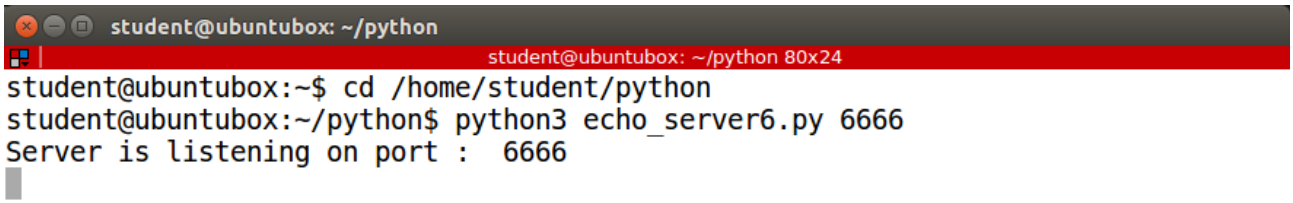
Server:

The server program can be run from the command-line with or without arguments:
If it runs without argument, it will prompt the user to enter a port number.

Without arguments:



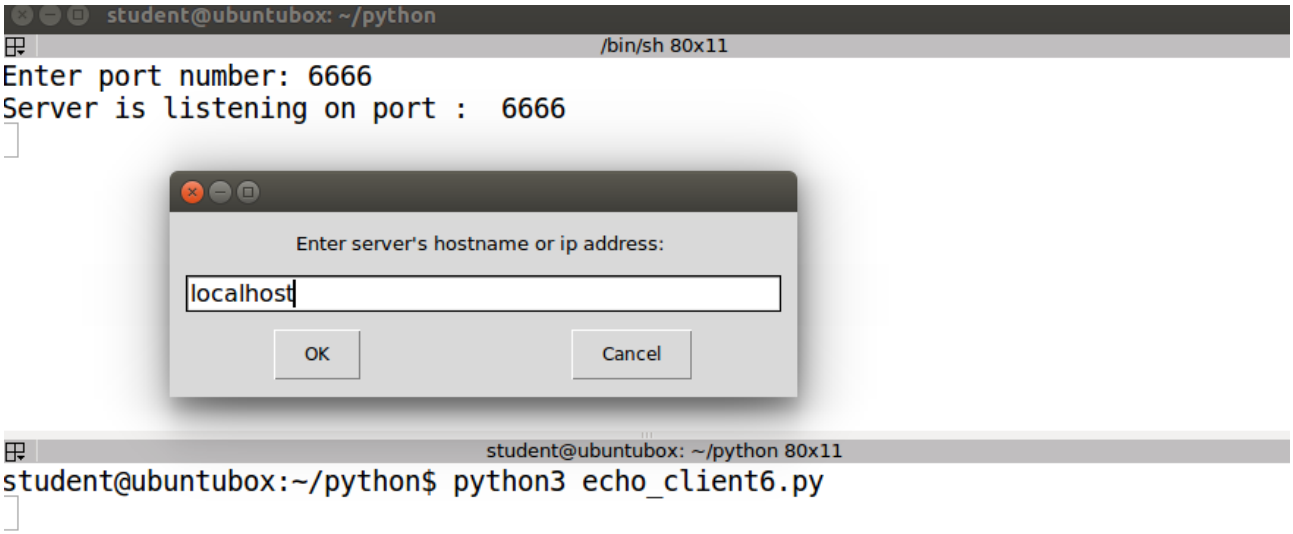
With argument specified port number:



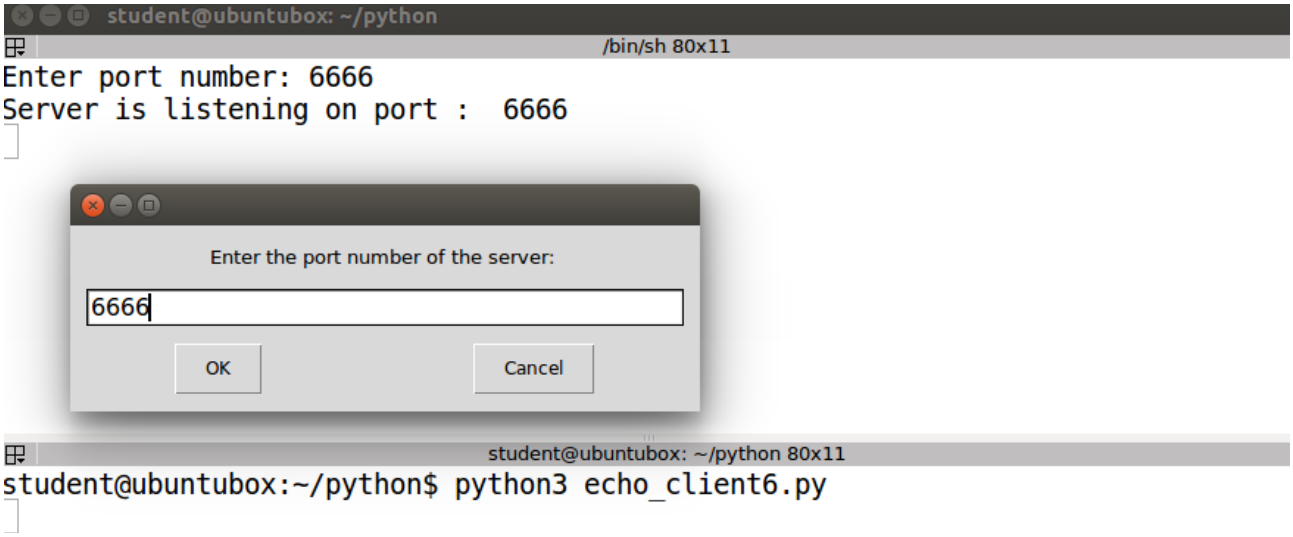
Client:

The client program can be run from the command-line with and without arguments.
If run without arguments, it will prompt the user to enter the server ip address and the server port number.

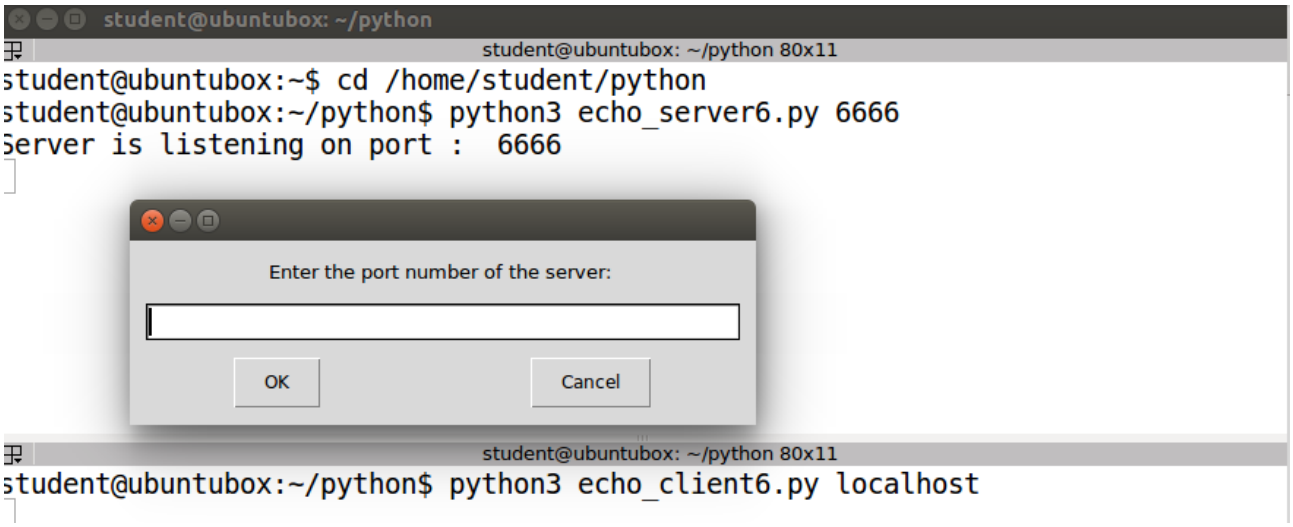
Without any of the arguments:
First enter the hostname:



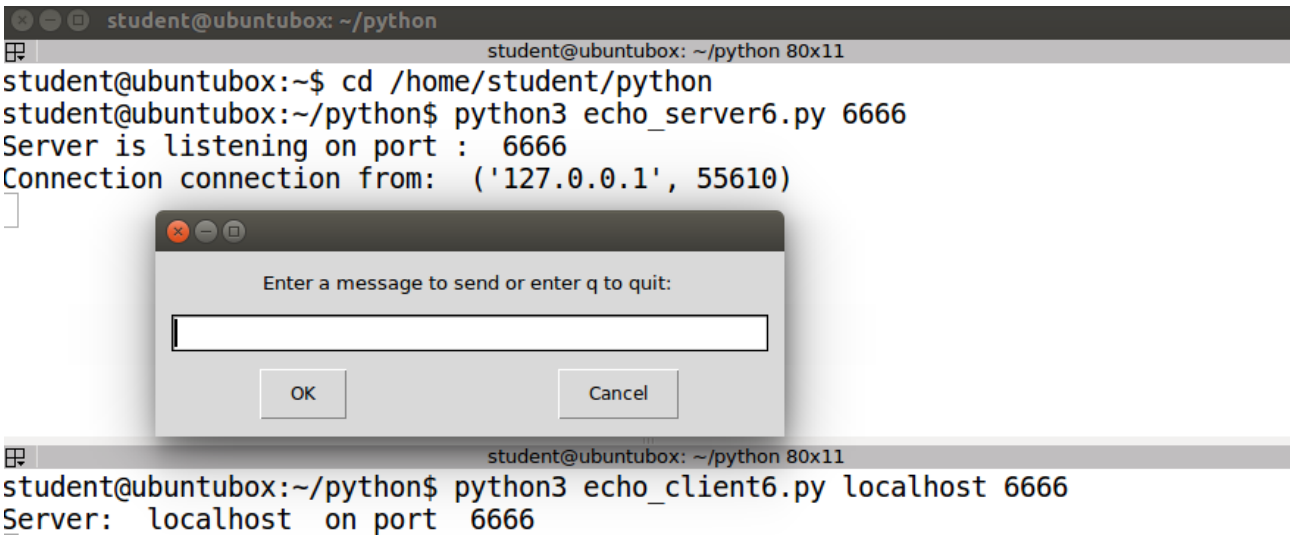
Then, enter the port number:



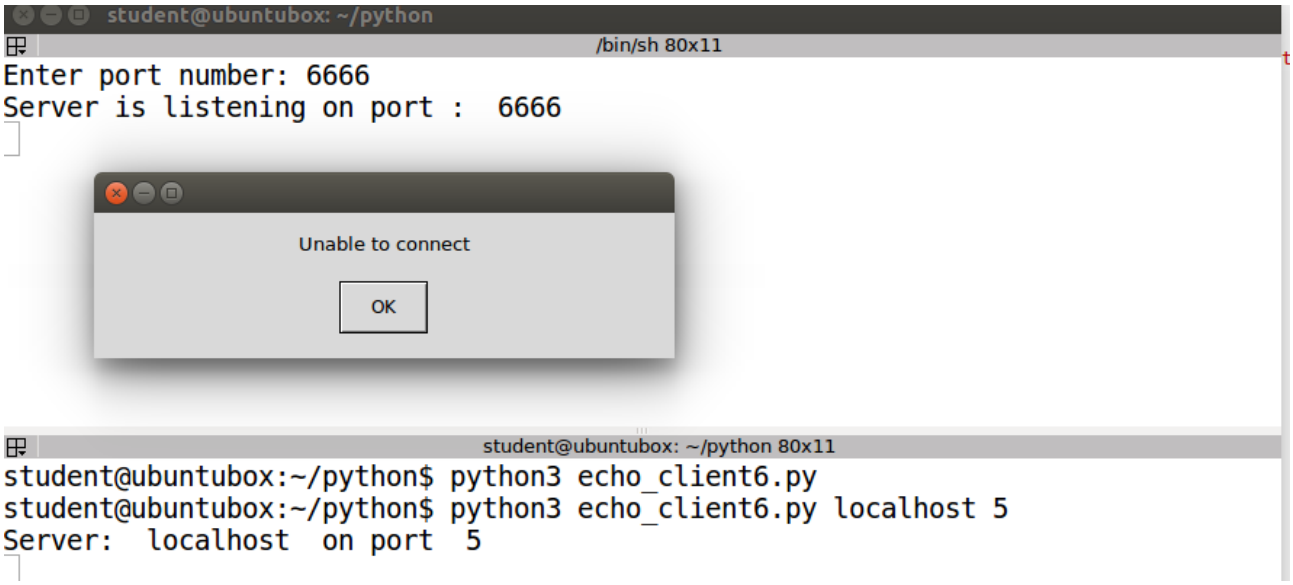
With only hostname argument:
Program will prompt you to enter the port number

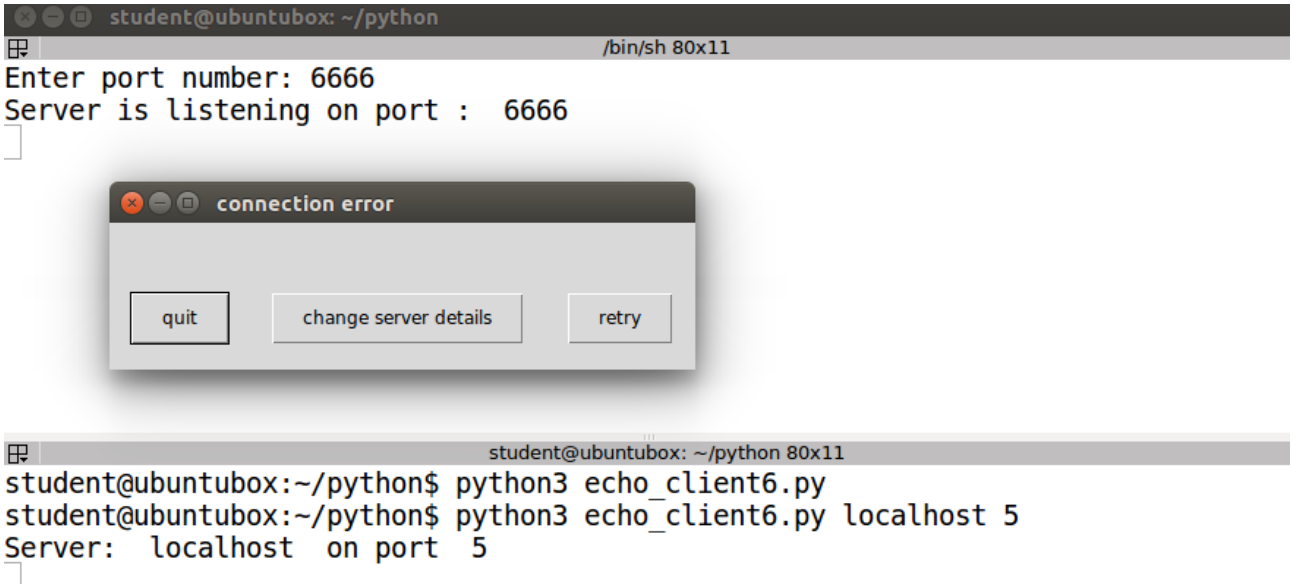


With arguments:

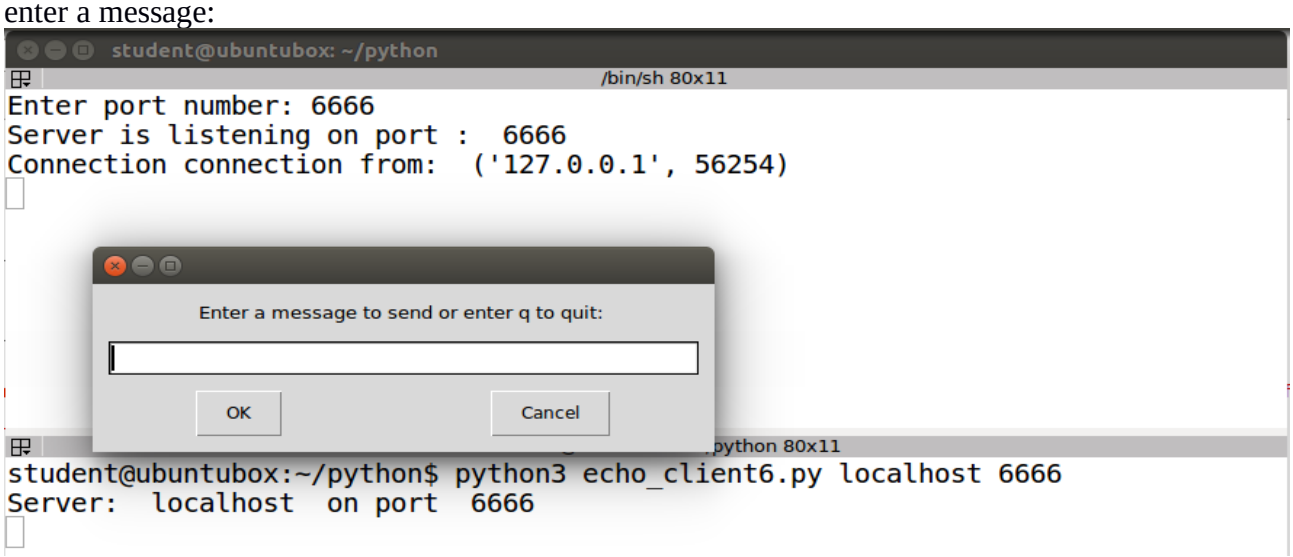


If the user enters a wrong information, first error message will be displayed on the screen and then program will give you 3 options: exit, retry, change server details

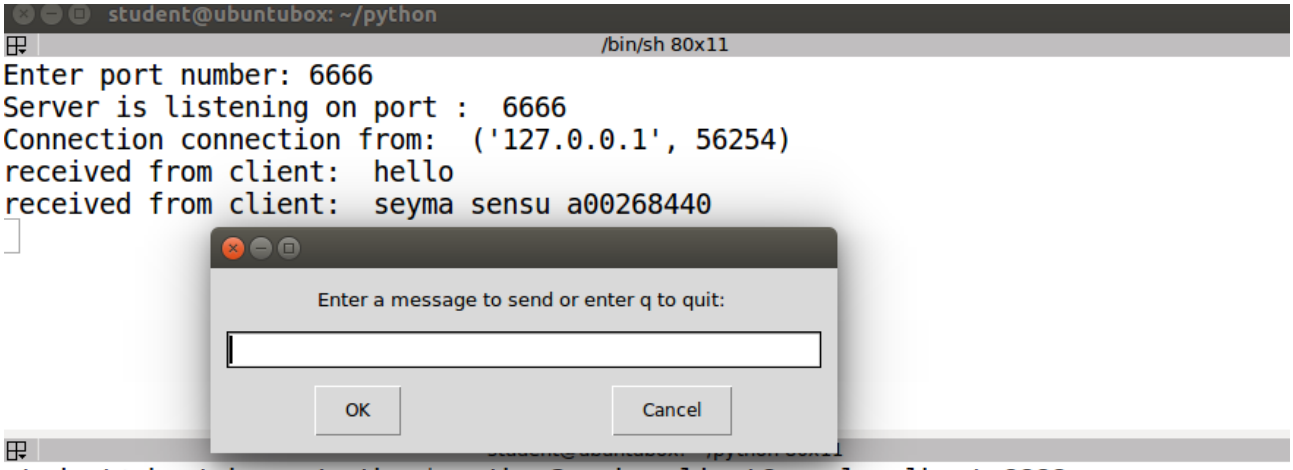




If the user enters the correct information a messagebox will be displayed and user will be asked to enter a message:

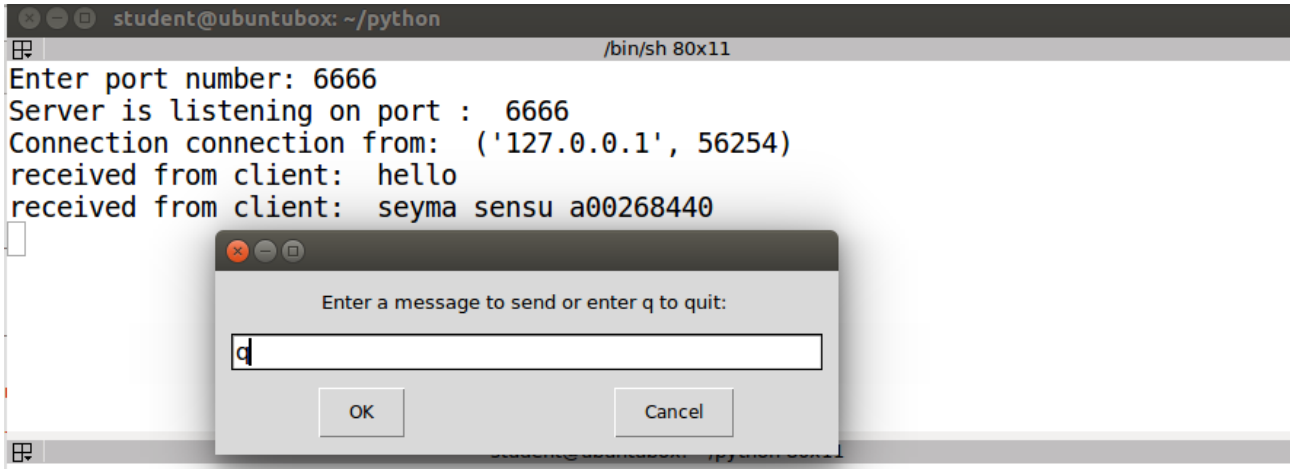


Messages entered by the user:
Program displays both the original message and hashed message.

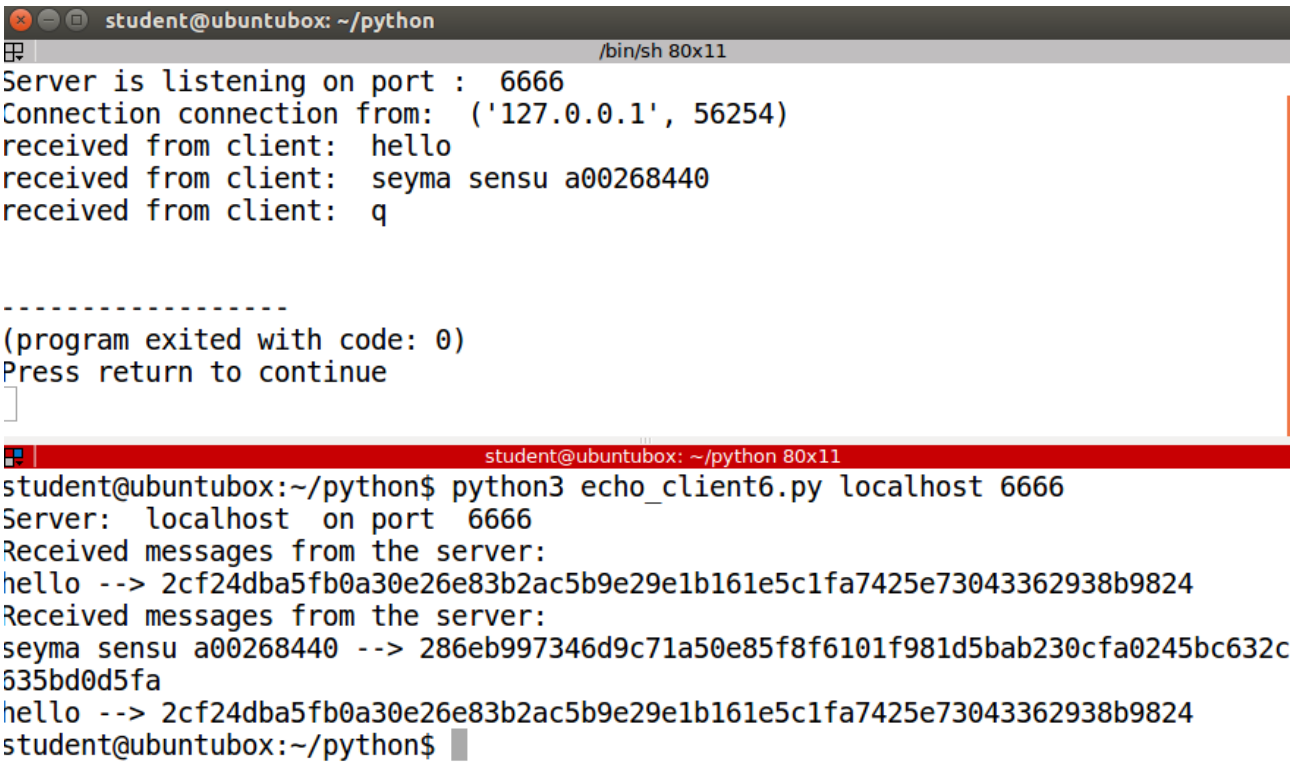



```
student@ubuntubox:~/python$ python3 echo_client6.py localhost 6666
Server:  localhost  on port  6666
Received messages from the server:
hello --> 2cf24dba5fb0a30e26e83b2ac5b9e29e1b161e5c1fa7425e73043362938b9824
Received messages from the server:
seyma sensu a00268440 --> 286eb997346d9c71a50e85f8f6101f981d5bab230cfa0245bc632c
635bd0d5fa
hello --> 2cf24dba5fb0a30e26e83b2ac5b9e29e1b161e5c1fa7425e73043362938b9824
```

If the user enters q program will be terminated:




```
student@ubuntubox:~/python$ python3 echo_client6.py localhost 6666
Server:  localhost  on port  6666
Received messages from the server:
hello --> 2cf24dba5fb0a30e26e83b2ac5b9e29e1b161e5c1fa7425e73043362938b9824
Received messages from the server:
seyma sensu a00268440 --> 286eb997346d9c71a50e85f8f6101f981d5bab230cfa0245bc632c
635bd0d5fa
hello --> 2cf24dba5fb0a30e26e83b2ac5b9e29e1b161e5c1fa7425e73043362938b9824
```


student@ubuntubox: ~/python
/bin/sh 80x11
Server is listening on port : 6666
Connection connection from: ('127.0.0.1', 56254)
received from client: hello
received from client: seyma sensu a00268440
received from client: q

(program exited with code: 0)
Press return to continue
student@ubuntubox: ~/python 80x11
student@ubuntubox:~/python$ python3 echo_client6.py localhost 6666
Server: localhost on port 6666
Received messages from the server:
hello --> 2cf24dba5fb0a30e26e83b2ac5b9e29e1b161e5c1fa7425e73043362938b9824
Received messages from the server:
seyma sensu a00268440 --> 286eb997346d9c71a50e85f8f6101f981d5bab230cfa0245bc632c
635bd0d5fa
hello --> 2cf24dba5fb0a30e26e83b2ac5b9e29e1b161e5c1fa7425e73043362938b9824
student@ubuntubox:~/python$

References

https://docs.python.org/3/library/hashlib.html

Reference Sheets on Moodle