Lab 5

**Submitted By:**

**Sana Naz**

**197631**

**BSCS-6B**

## Server Code:

# import socket programming library

import socket

# import thread module

from \_thread import \*

import threading

import pickle

print\_lock = threading.Lock()

my\_dict = {

"brand": "Ford",

"model": "Mustang",

"year": "1964"

}

# thread fuction

def threaded(c):

while True:

# data received from client

data = pickle.loads(c.recv(1024))

if not data:

print('ERROR: Input was not received.')

c.send(message.encode('ascii'))

elif (data["option"] == "1"):

if data["word"] not in my\_dict:

my\_dict[data["word"]] = data["meaning"]

message = "Word added in dictionary."

c.send(message.encode('ascii'))

else:

message = "Word already added in dictionary."

c.send(message.encode('ascii'))

elif(data["option"] == "2"):

if data["word"] in my\_dict:

message = "Meaning: "+my\_dict[data["word"]]

c.send(message.encode('ascii'))

else:

message = "The word you are looking for is not present in the dictionary."

c.send(message.encode('ascii'))

elif(data["option"] == "3"):

if data["word"] in my\_dict:

my\_dict.pop(data["word"])

message = "Word deleted from dictionary."

c.send(message.encode('ascii'))

else:

message = "The word you are trying to delete is not present in the dictionary."

c.send(message.encode('ascii'))

else:

message = "ERROR: Incorrect option."

c.send(message.encode('ascii'))

#print("Message: ",message)

#c.send(message.encode('ascii'))

# lock released on exit

print\_lock.release()

break

# connection closed

c.close()

def Main():

host = ""

# reverse a port on your computer

# in our case it is 12345 but it

# can be anything

port = 12345

s = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)

s.bind((host, port))

print("socket binded to port", port)

# put the socket into listening mode

s.listen(5)

print("socket is listening")

# a forever loop until client wants to exit

while True:

# establish connection with client

c, addr = s.accept()

# lock acquired by client

print\_lock.acquire()

print('Connected to :', addr[0], ':', addr[1])

# Start a new thread and return its identifier

start\_new\_thread(threaded, (c,))

s.close()

if \_\_name\_\_ == '\_\_main\_\_':

Main()

## Server Output:



## Client Code:

## # Import socket module

## import socket

## import pickle

## 

## def Main():

## # local host IP '127.0.0.1'

## host = '127.0.0.1'

## 

## # Define the port on which you want to connect

## port = 12345

## 

## s = socket.socket(socket.AF\_INET,socket.SOCK\_STREAM)

## 

## # connect to server on local computer

## s.connect((host,port))

## 

## # message you send to server

## message\_dict = {}

## 

## while True:

## option = input("\nPlease select an option. \nEnter 1 to add a new word to the dictionary.\nEnter 2 to search for a word.\nEnter 3 to delete a word.\nOption No: ")

## if(option == "1"):

## print("\nAdding new word to dictionary\n")

## message\_dict["option"] = "1"

## message\_dict["word"] = input("Enter word: ")

## message\_dict["meaning"] = input("Enter meaning: ")

## s.send(pickle.dumps(message\_dict,-1))

## 

## elif(option == "2"):

## print("\nSearch word from dictionary\n")

## message\_dict["option"] = "2"

## message\_dict["word"] = input("Enter word: ")

## s.send(pickle.dumps(message\_dict,-1))

## 

## elif(option == "3"):

## print("\nDelete word from dictionary\n")

## message\_dict["option"] = "3"

## message\_dict["word"] = input("Enter word: ")

## s.send(pickle.dumps(message\_dict,-1))

## 

## else:

## print("ERROR: The option you entered is not valid. Please try again.\n")

## continue

## 

## # messaga received from server

## data = s.recv(1024)

## 

## # print the received message

## print('Received from the server :',str(data.decode('ascii')))

## 

## # ask the client whether he wants to continue

## ans = input('\nDo you want to continue(y/n) :')

## if ans == 'y':

## continue

## else:

## break

## # close the connection

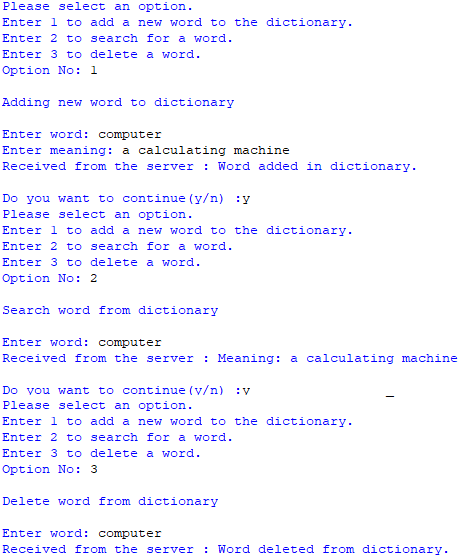
## s.close()

## 

## if \_\_name\_\_ == '\_\_main\_\_':

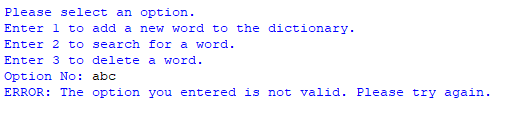
## Main()

## Client Output:

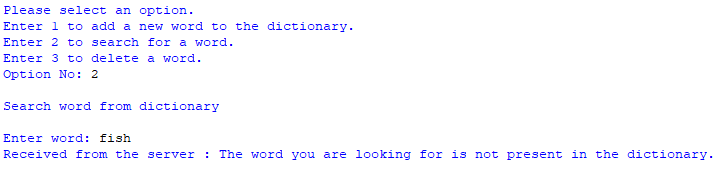


## Errors:

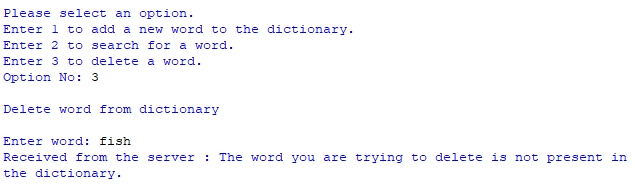
### Invalid option entered



### Word we are searching for is not in dictionary:



### Word we are trying to delete is not in dictionary



### Word already added to dictionary

