ASSIGNMENT -1

Date: 16.01.2019

Topic: Object Serialization and Deserialization

1. Design a Student class with data field name and age. Then, create two objects of this student class named as Student1 and Student2. Pass the values name=Sita and age = 23 to Student1 and name=Gita and age=24 to Student2. Apply the concept of serialization and store the value of Student1 and Student2 objects in 2 different text files. After this apply the concept of deserialization to retrieve the stored states of Student1 and Student2 objects.

Solution:

```
import pickle
class Student :
   def __init__(self, name, age):
         self.name = name
         self.age = age

Student1 = Student("Sita",23)
Student2 = Student("Gita",24)

print (str.format("The details of student 1 are:\nName:{0}\tAge:{1}",Student1.name,
Student1.age))

print (str.format("\nThe details of student 2 are:\nName:{0}\tAge:{1}",Student2.name,
Student2.age))


Student1_file= open ('Student1.txt',mode='wb')
Student1_pickled = pickle.dump(Student1, Student1_file)
Student1_file.close()

Student2_file= open ('Student2.txt',mode='wb')
Student2_pickled = pickle.dump(Student2, Student2_file)
Student2_file.close()
```

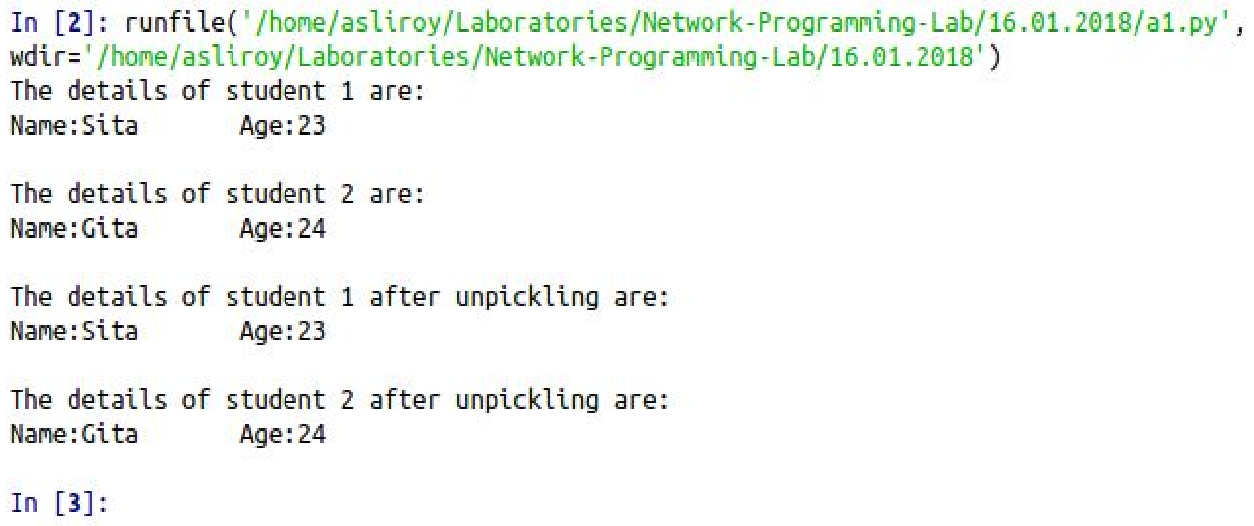
```
Student1_file= open ('Student1.txt',mode='rb')
Student1_unpickled = pickle.load(Student1_file)
Student1_file.close()

Student2_file= open ('Student2.txt',mode='rb')
Student2_unpickled = pickle.load(Student2_file)
Student2_file.close()


print(str.format("\nThe details of student 1 after unpickling
are:\nName:{0}\tAge:{1}",Student1_unpickled.name, Student1_unpickled.age))


print(str.format("\nThe details of student 2 after unpickling
are:\nName:{0}\tAge:{1}",Student2_unpickled.name, Student2_unpickled.age))
```

Output:



```
In [2]: runfile('/home/asliroy/Laboratories/Network-Programming-Lab/16.01.2018/a1.py',
wdir='/home/asliroy/Laboratories/Network-Programming-Lab/16.01.2018')
The details of student 1 are:
Name:Sita       Age:23

The details of student 2 are:
Name:Gita       Age:24

The details of student 1 after unpickling are:
Name:Sita       Age:23

The details of student 2 after unpickling are:
Name:Gita       Age:24

In [3]:
```

ASSIGNMENT -2

Date: 23.01.2019

Topic: Application Layer Protocol

1. Write a Python network server program that will accept an unlimited number of connections, one at a time. Upon receiving a connection, it should send back to the client, the client's Ip address. Then it should wait for commands from the client. Valid commands are 'TIME', 'IP','EXIT'. To the time command, the server should return the current time. To the IP command, it should again return the client's IP address. If the client closes the connection or does not respond within a reasonable time (10 seconds), the server should close the current connection and wait for another connection. to the EXIT command, your server should close all open sockets and exit.

Solution:

Client side code:

```
import socket,sys

s = socket.socket()
host = socket.gethostname()
port = 12346

s.connect((host, port))
print s.recv(1024)
shell = sys.stdin.readline().strip()
s.send(shell)
data = s.recv(1024)
output = data
print 'Received', repr(data)

s.close()
```

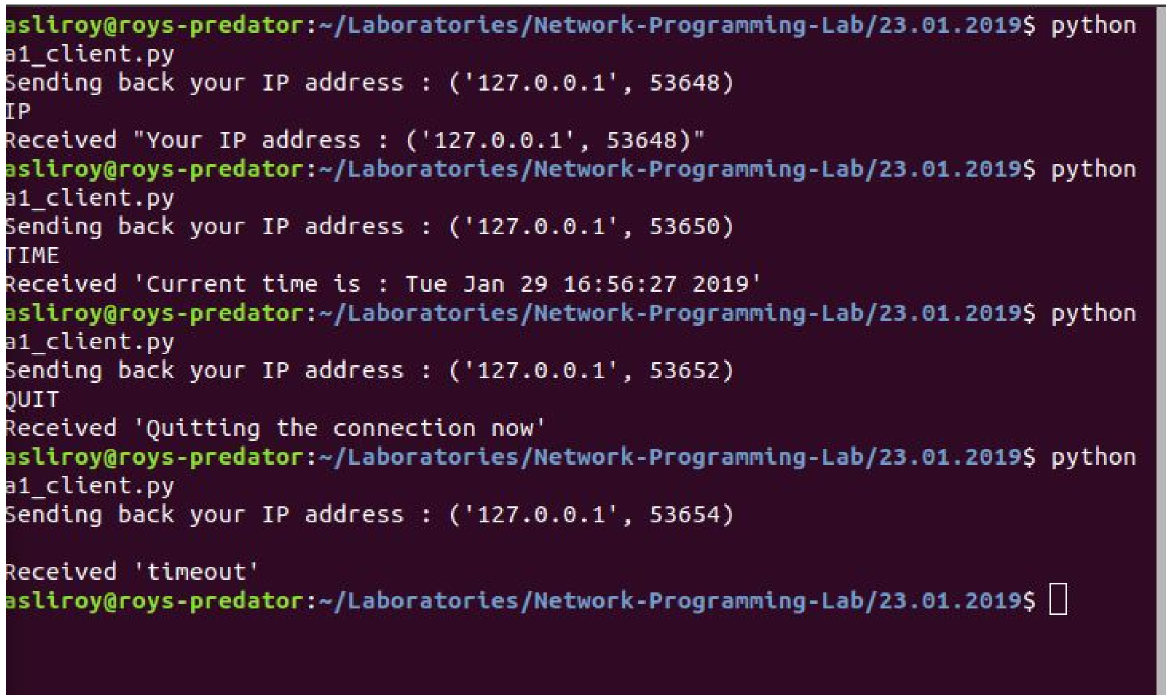
Server side code:

```
import socket
import time
```

```
host = socket.gethostname()
port = 12346
socksize = 1024
s = socket.socket()
s.bind((host, port))
print(" The server started on port: %s" %port)
s.listen(1)
print("The server is now listening for potential clients->->->")
while True:
    c, addr = s.accept()
    print 'New client connection accepted from %s:%d' % (addr[0], addr[1])
    c.settimeout(10)
    c.send('Sending back your IP address : {}'.format(addr))

    try:
        data = c.recv(socksize)
        if data == 'QUIT':
            c.send('Quitting the connection now')
            print ("Connection has been terminated by the client.")
            c.close()
        elif data == 'IP':
            c.send('Your IP address : {}'.format(addr))
        elif data == 'TIME':
            localtime = time.asctime( time.localtime(time.time()) )
            t = localtime
            c.send('Current time is : {}'.format(t))
    except socket.timeout:
        print ("10 second timeout, please retry again")
        c.send('timeout')
        c.close()
```

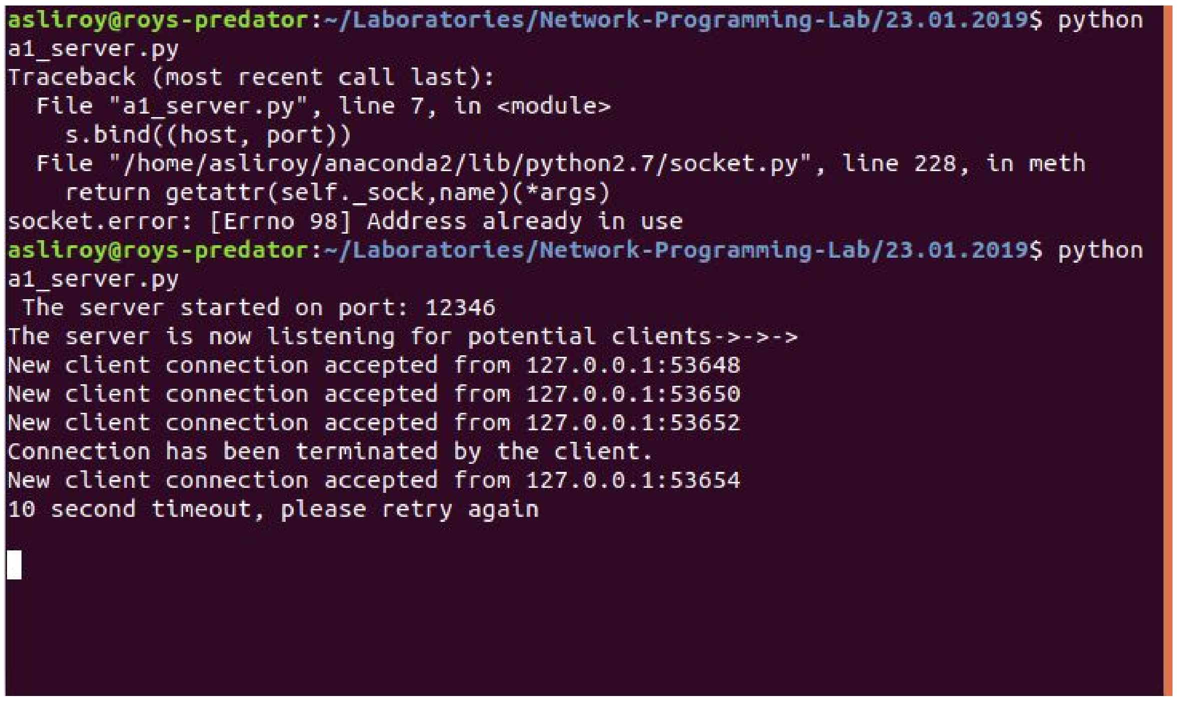
Output:



```
asliroy@roys-predator:~/Laboratories/Network-Programming-Lab/23.01.2019$ python
a1_client.py
Sending back your IP address : ('127.0.0.1', 53648)
IP
Received "Your IP address : ('127.0.0.1', 53648)"
asliroy@roys-predator:~/Laboratories/Network-Programming-Lab/23.01.2019$ python
a1_client.py
Sending back your IP address : ('127.0.0.1', 53650)
TIME
Received 'Current time is : Tue Jan 29 16:56:27 2019'
asliroy@roys-predator:~/Laboratories/Network-Programming-Lab/23.01.2019$ python
a1_client.py
Sending back your IP address : ('127.0.0.1', 53652)
QUIT
Received 'Quitting the connection now'
asliroy@roys-predator:~/Laboratories/Network-Programming-Lab/23.01.2019$ python
a1_client.py
Sending back your IP address : ('127.0.0.1', 53654)

Received 'timeout'
asliroy@roys-predator:~/Laboratories/Network-Programming-Lab/23.01.2019$
```

Client Output



```
asliroy@roys-predator:~/Laboratories/Network-Programming-Lab/23.01.2019$ python
a1_server.py
Traceback (most recent call last):
  File "a1_server.py", line 7, in <module>
    s.bind((host, port))
  File "/home/asliroy/anaconda2/lib/python2.7/socket.py", line 228, in meth
    return getattr(self._sock,name)(*args)
socket.error: [Errno 98] Address already in use
asliroy@roys-predator:~/Laboratories/Network-Programming-Lab/23.01.2019$ python
a1_server.py
 The server started on port: 12346
The server is now listening for potential clients->->->
New client connection accepted from 127.0.0.1:53648
New client connection accepted from 127.0.0.1:53650
New client connection accepted from 127.0.0.1:53652
Connection has been terminated by the client.
New client connection accepted from 127.0.0.1:53654
10 second timeout, please retry again
```

Server Output