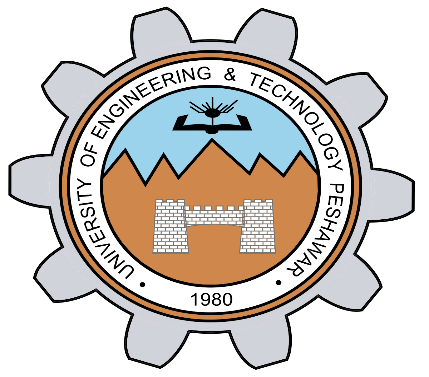
# Lab 09

**TCP/IP Implementation using Python Socket Programming**



**CSE 303L: Data Communication and Computer Networks**

Submitted by: Ashfaq Ahmad

Registration No. : 19pwcse1795

Class Section: **B**

Submitted to:

**Engr. FaizUllah**

Department of Computer Systems Engineering

University of Engineering and Technology, Peshawar

**CSE 303L: Data Communication and Computer Networks**

**Credit Hours: 1**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Demonstration of Concepts** | **Poor (Does not meet expectation (1))**  The student failed to demonstrate a clear understanding of the assignment concepts | **Fair (Meet Expectation (2-3))**  The student demonstrated a clear understanding of some of the assignment concepts | **Good (Exceeds Expectation (4-5)**  The student demonstrated a clear understanding of the assignment concepts | **Score**  **30%** |
| **Accuracy** | The student mis-configured enough network settings that the lab computer couldn't function properly on the network | The student configured enough network settings that the lab computer partially functioned on the network | The student configured the network settings that the lab computer fully functioned on the network | **30%** |
| **Following Directions** | The student clearly failed to follow the verbal and written instructions to successfully complete the lab | The student failed to follow the some of the verbal and written instructions to successfully complete all requirements of the lab | The student followed the verbal and written instructions to successfully complete requirements of the lab | **20%** |
| **Time Utilization** | The student failed to complete even part of the lab in the allotted amount of time | The student failed to complete the entire lab in the allotted amount of time | The student completed the lab in its entirety in the allotted amount of time | **20%** |

**LAB # 9**

**TCP/IP Implementation using Python Socket Programming**

|  |  |  |
| --- | --- | --- |
| **Criteria** | **Excellent** | **Marks Obtained** |
| 1. **Objectives of Lab** | All objectives of lab are properly covered  [Marks 0.5] |  |
| 1. **Introduction to Python Programming** | Brief introduction of Python Programming  [Marks 2] |  |
| 1. **Introduction to python socket library and its various functions** | Brief introduction about Socket library and its various functions used in Lab  [Marks 2] |  |
| 1. **Client-Server Communication using socket library** | Client-Server communication, Python code and output  [Marks 3] |  |
| 1. **Flowchart of client server communication using python socket library** | Draw a flowchart of  the sequence of socket API calls and data flow for TCP  [Marks 2] |  |
| 1. **Conclusion** | Conclusion about RC-Circuit analysis  [Marks 0.5] |  |

**Objectives:**

Introduction to Python Programming

Introduction to Socket Library

Client-Server Communication

**Python Programming**

Python is a high-level, interpreted, general-purpose programming language. Its design philosophy emphasizes code readability with the use of significant indentation. Python is dynamically-typed and garbage-collected. It supports multiple programming paradigms, including structured, object-oriented and functional programming.

Tuple = (1,2,3,4)

List =[1,2,3,4]

Tuple is immutable while list is not. On runtime no value can be inserted

nor deleted from tuple but list can be updated during runtime.

Dictionary {key: value}

{'name':'Zain'}

for i in range(1,10,2):

#upper limit will not be included

#third arg reps step

if i==4 or i==5:

print("i=",i)

elif:

else:

i = 5

while i>1:

i-=1 #no ++i or --i

**Socket Library**

Sockets and the socket API are used to send messages across a network. They provide a form of inter-process communication (IPC). The network can be a logical, local network to the computer, or one that’s physically connected to an external network, with its own connections to other networks. The obvious example is the Internet, which you connect to via your ISP.

The primary socket API functions and methods are:

* socket()
* .bind()
* .listen()
* .accept()
* .connect()
* .connect\_ex()
* .send()
* .recv()
* .close()

**Server Code**

import socket

s=socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)#ipv4, tcp

s.bind((socket.gethostname(),6061)) #port-no = 6061

s.listen(5)

clientSocket, address = s.accept()

i = 1

while(i):

msg = input("Enter your message")

clientSocket.send(bytes(msg,'utf-8'))

clientSocket.close()

**Client Code**

import socket

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

s.connect((socket.gethostname(),6061))

i = 1

while(i):

msg = s.recv(2048)

print("Message recieved",msg)

