

1. If you run a TCP client without the server, there will be an error, because there was never a connection established with the server.
2. Localhost is the computer or hostname that is currently in use to run a program and serves as a virtual server. It is used to develop networked applications because it allows developers to test their applications without a connection to the web to do so.
3. The `time.sleep()` function adds a delay in the execution of a program.
4. a. `bind()`- connects a port number and IP address to a socket
 - a. `connect()` - connects a TCP-based client socket to a TCP-based server socket.
 - b. `send()`- sends data from one socket to another
 - c. `recv()` - receive messages from another socket
 - d. `accept()` - accepts an incoming connection request from a TCP client.

5

The first screenshot shows the server code in `sockserver.py`. It imports `socket`, creates a `socket` object, binds it to `localhost` on port `5000`, and starts listening. A `while` loop accepts connections, receives data, prints it, and sends back a response before closing the connection.

```
1 import socket
2 socsocket = socket()
3 print("Socket created successfully")
4 soc.bind(("localhost", 5000))
5
6 soc.listen(4)
7
8 while True:
9     conn, addr = soc.accept()
10    message = conn.recv(1024).decode()
11    print(message)
12    print("Got connection from", addr)
13    conn.send("Hey! I am the server!".encode())
14    conn.send("Welcome all!".encode())
15    conn.close()
```

The second screenshot shows the client code in `sockclient.py`. It imports `socket`, creates a `socket` object, connects it to `localhost` on port `5000`, sends a message, receives a response, and prints it before closing the connection.

```
1 import socket
2 socsocket = socket()
3
4 soc.connect(("localhost", 5000))
5 message = "I am the client"
6 soc.send(message.encode())
7 resp1 = soc.recv(1024).decode()
8 print(resp1)
9 resp2 = soc.recv(1024).decode()
10 print(resp2)
11
12 soc.close()
```

Both screenshots show the Run console output, indicating that the server and client are running successfully and communicating.

The screenshot shows the PyCharm IDE interface. The main editor window displays a Python script named `client2.py` with the following code:

```
1 import socket
2 socksocket.socket()
3
4 soc.connect(("localhost", 5000))
5 message = "I am the second client"
6 soc.send(message.encode())
7 resp = soc.recv(1024).decode()
8 print(resp)
9 resp2 = soc.recv(1024).decode()
10 print(resp2)
11
12 soc.close()
```

Below the editor, the 'Run' console shows the execution output for the `client2` configuration. The output indicates that the process finished successfully with exit code 0, and the messages received from the server are:

```
"C:\Users\Lee's PC\PycharmProjects\pythonProject\HelloWorld\venv\Scripts\python.exe" "C:/Users/Lee's PC/PycharmProjects/pythonPro
Hey! I am the server!
Welcome all!
Process finished with exit code 0
```