13.Write a client and server programs using TCP and UDP.

Theory:

Java Networking is a concept of connecting two or more computing devices together so that we can share resources.

Java socket programming provides facility to share data between different computing devices.

The java.net package supports two protocols,

1.**TCP:** Transmission Control Protocol provides reliable communication between the sender and receiver. TCP is used along with the Internet Protocol referred as TCP/IP.

2.**UDP:** User Data-gram Protocol provides a connection-less protocol service by allowing packet of data to be transferred along two or more nodes

Source Code For TCP Socket:

File: SocketServerExample.java

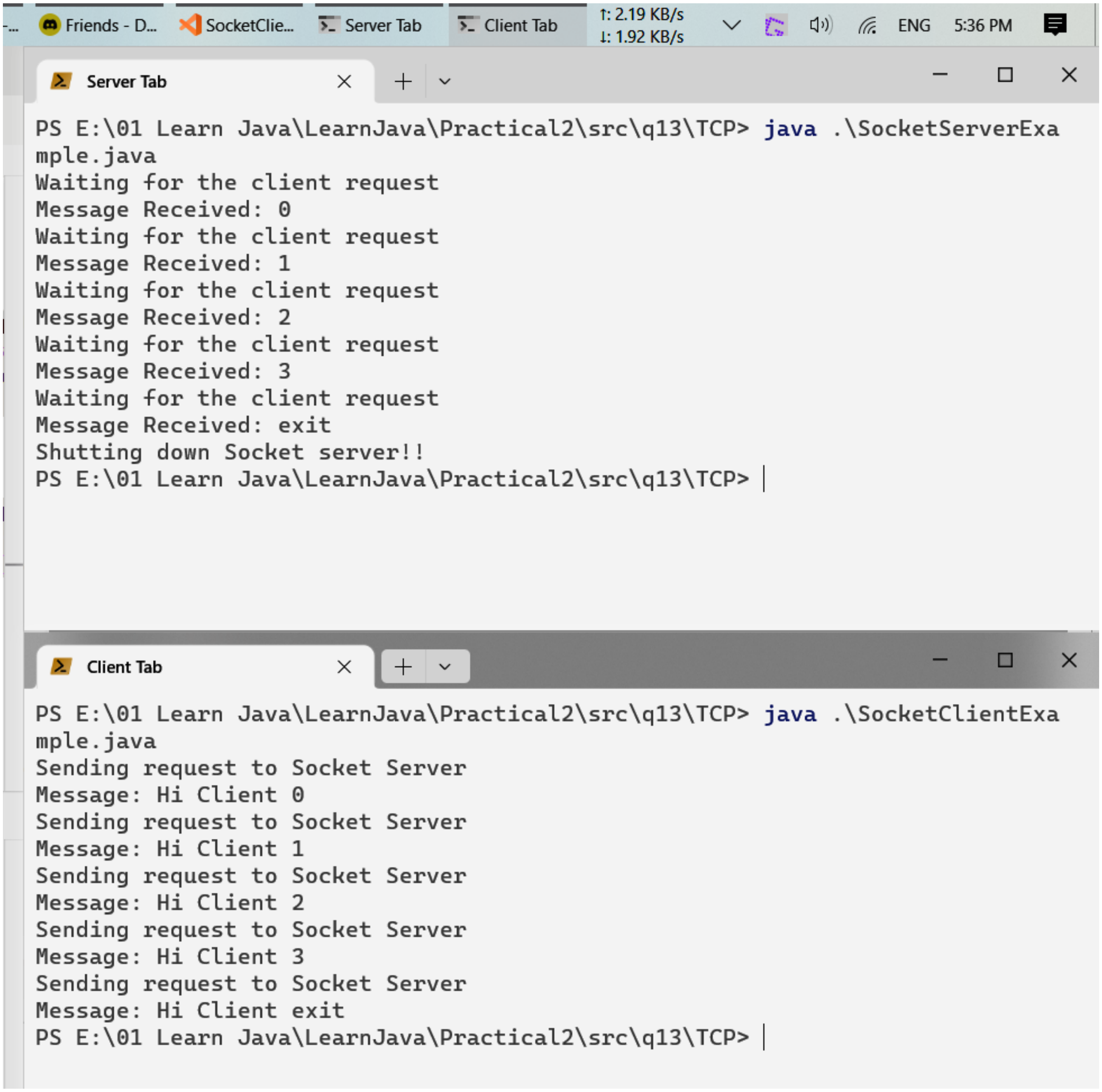
```
import java.io.IOException;
import java.io.ObjectInputStream;
import java.io.ObjectOutputStream;
import java.lang.ClassNotFoundException;
import java.net.ServerSocket;
import java.net.Socket;
public class SocketServerExample {
    private static ServerSocket server;
    private static int port = 9876;
    public static void main(String args[]) throws IOException, ClassNotFoundException{
        server = new ServerSocket(port);
        while(true){
            System.out.println("Waiting for the client request");
            Socket socket = server.accept();
            ObjectInputStream ois = new ObjectInputStream(socket.getInputStream());
            String message = (String) ois.readObject();
            System.out.println("Message Received: " + message);
            ObjectOutputStream oos = new ObjectOutputStream(socket.getOutputStream());
                  oos.writeObject("Hi Client "+message);
            ois.close();
      oos.close();
      socket.close();
            if(message.equalsIgnoreCase("exit")) break;
    }
    System.out.println("Shutting down Socket server!!");
        server.close();
  }

}
```

File: SocketClientExample.java

```
import java.io.IOException;
import java.io.ObjectInputStream;
import java.io.ObjectOutputStream;
import java.net.InetAddress;
import java.net.Socket;
import java.net.UnknownHostException;
public class SocketClientExample {
    public static void main(String[] args) throws UnknownHostException, IOException,
ClassNotFoundException, InterruptedException{
                InetAddress host = InetAddress.getLocalHost();
        Socket socket = null;
        ObjectOutputStream oos = null;
        ObjectInputStream ois = null;
        for(int i=0; i<5;i++){
                        socket = new Socket(host.getHostName(), 9876);
                        oos = new ObjectOutputStream(socket.getOutputStream());
            System.out.println("Sending request to Socket Server");
            if(i==4)oos.writeObject("exit");
            else oos.writeObject(""+i);
                        ois = new ObjectInputStream(socket.getInputStream());
            String message = (String) ois.readObject();
            System.out.println("Message: " + message);
                        ois.close();
            oos.close();
            Thread.sleep(100);
        }
    }
}
```

Output:



```
PS E:\01 Learn Java\LearnJava\Practical2\src\q13\TCP> java .\SocketServerExa
mple.java
Waiting for the client request
Message Received: 0
Waiting for the client request
Message Received: 1
Waiting for the client request
Message Received: 2
Waiting for the client request
Message Received: 3
Waiting for the client request
Message Received: exit
Shutting down Socket server!!
PS E:\01 Learn Java\LearnJava\Practical2\src\q13\TCP>
```

```
PS E:\01 Learn Java\LearnJava\Practical2\src\q13\TCP> java .\SocketClientExa
mple.java
Sending request to Socket Server
Message: Hi Client 0
Sending request to Socket Server
Message: Hi Client 1
Sending request to Socket Server
Message: Hi Client 2
Sending request to Socket Server
Message: Hi Client 3
Sending request to Socket Server
Message: Hi Client exit
PS E:\01 Learn Java\LearnJava\Practical2\src\q13\TCP>
```