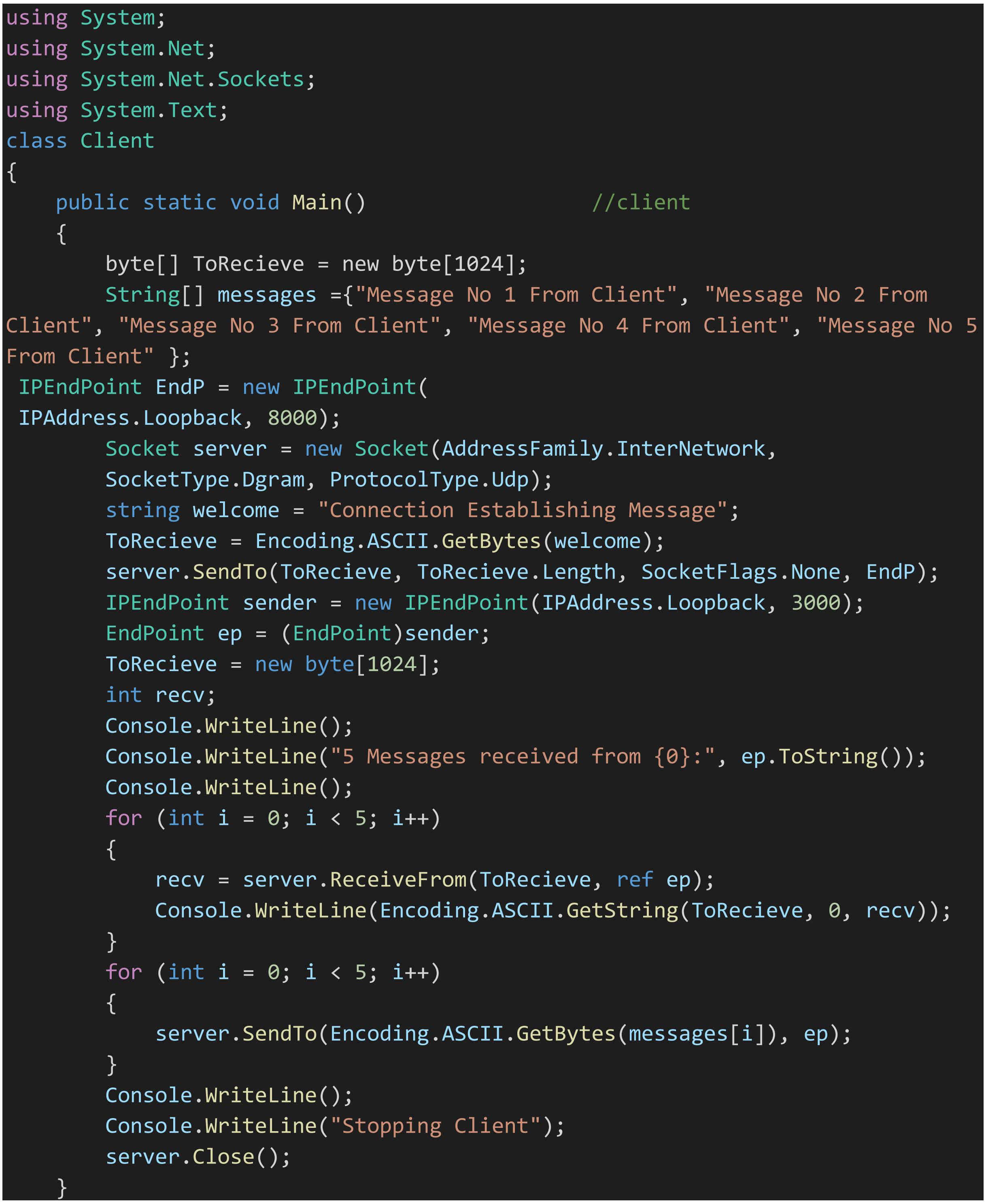
MUHAMMAD UMAR KHAN 10619
NETWORK PROGRAMMING 108157
ASSIGNMENT#02

Q1.

<u>CLIENT:</u>

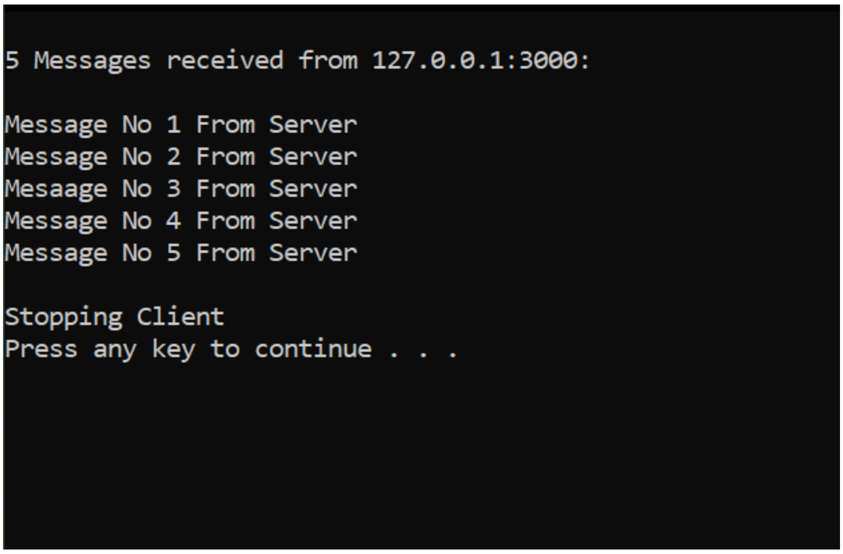


```
using System;
using System.Net;
using System.Net.Sockets;
using System.Text;
class Client
{
    public static void Main()                  //client
    {
        byte[] ToRecieve = new byte[1024];
        String[] messages ={"Message No 1 From Client", "Message No 2 From
Client", "Message No 3 From Client", "Message No 4 From Client", "Message No 5
From Client" };
 IPEndPoint EndP = new IPEndPoint(
 IPAddress.Loopback, 8000);
        Socket server = new Socket(AddressFamily.InterNetwork,
        SocketType.Dgram, ProtocolType.Udp);
        string welcome = "Connection Establishing Message";
        ToRecieve = Encoding.ASCII.GetBytes(welcome);
        server.SendTo(ToRecieve, ToRecieve.Length, SocketFlags.None, EndP);
        IPEndPoint sender = new IPEndPoint(IPAddress.Loopback, 3000);
        EndPoint ep = (EndPoint)sender;
        ToRecieve = new byte[1024];
        int recv;
        Console.WriteLine();
        Console.WriteLine("5 Messages received from {0}:", ep.ToString());
        Console.WriteLine();
        for (int i = 0; i < 5; i++)
        {
            recv = server.ReceiveFrom(ToRecieve, ref ep);
            Console.WriteLine(Encoding.ASCII.GetString(ToRecieve, 0, recv));
        }
        for (int i = 0; i < 5; i++)
        {
            server.SendTo(Encoding.ASCII.GetBytes(messages[i]), ep);
        }
        Console.WriteLine();
        Console.WriteLine("Stopping Client");
        server.Close();
    }
}
```

SERVER:

```
using System;
using System.Net;
using System.Net.Sockets;
using System.Text;
class Server
{
    public static void Main()           //server
    {
        int OutPut;
        byte[] ToRecieve = new byte[1024];
        String[] messages = { "Message No 1 From Server", "Message No 2 From
Server", "Mesaage No 3 From Server", "Message No 4 From Server", "Message No 5
From Server" };
 IPEndPoint EndP = new IPEndPoint(IPAddress.Loopback, 8000);
        Socket server = new Socket(AddressFamily.InterNetwork,
SocketType.Dgram, ProtocolType.Udp);
        server.Bind(EndP);
        Console.WriteLine("Listening To Client Requests");
        Console.WriteLine(" ");
        IPEndPoint sk = new IPEndPoint(IPAddress.Loopback, 3000);
        EndPoint ep = (EndPoint)(sk);
        OutPut = server.ReceiveFrom(ToRecieve, ref ep);
        Console.WriteLine(Encoding.ASCII.GetString(ToRecieve, 0, OutPut));
        for (int i = 0; i < 5; i++)
        {
            server.SendTo(Encoding.ASCII.GetBytes(messages[i]), ep);
        }
        Console.WriteLine();
        Console.WriteLine("5 Messages received from {0}:", ep.ToString());
        Console.WriteLine();
        for (int i = 0; i < 5; i++)
        {
            ToRecieve = new byte[1024];
            OutPut = server.ReceiveFrom(ToRecieve, ref ep);
            Console.WriteLine(Encoding.ASCII.GetString(ToRecieve, 0, OutPut));
        }
        server.Close();
    }
}
```

OUTPUT:



```
5 Messages received from 127.0.0.1:3000:

Message No 1 From Server
Message No 2 From Server
Mesaage No 3 From Server
Message No 4 From Server
Message No 5 From Server

Stopping Client
Press any key to continue . . .
```

Q2.

SERVER:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Net;
using System.Net.Sockets;
using System.Threading;
namespace Server
{
    class Server    //SERVER
    {
        static void Main(string[] args)
        {

            byte[] data = new byte[1024];
            IPEndPoint Endp = new IPEndPoint(IPAddress.Loopback, 3000);
            Socket server = new Socket(AddressFamily.InterNetwork,
SocketType.Dgram, ProtocolType.Udp);
            server.Bind(Endp);
            Console.WriteLine("Waiting For Clients");
            Server Serverobj = new Server();
            while (true)
            {
                ThreadStart StartClient = new ThreadStart(() =>
Serverobj.Client(server));
                Thread th = new Thread(StartClient);
                th.Start();
            }
        }
        public void Client(Socket s)
        {

            while (true)
            {
                IPEndPoint sender = new IPEndPoint(IPAddress.Any, 0);
                EndPoint ep = (EndPoint)(sender);
                byte[] ToReceive = new byte[1024];
                int output = s.ReceiveFrom(ToReceive, ref ep);
                Console.WriteLine("Message From Client");
                Console.WriteLine();
Console.WriteLine(Encoding.ASCII.GetString(ToReceive, 0, output));
                string Res = "Response Message From Server To Client";
                s.SendTo(Encoding.ASCII.GetBytes(Res), ep);
            }
        }
    }
}
```

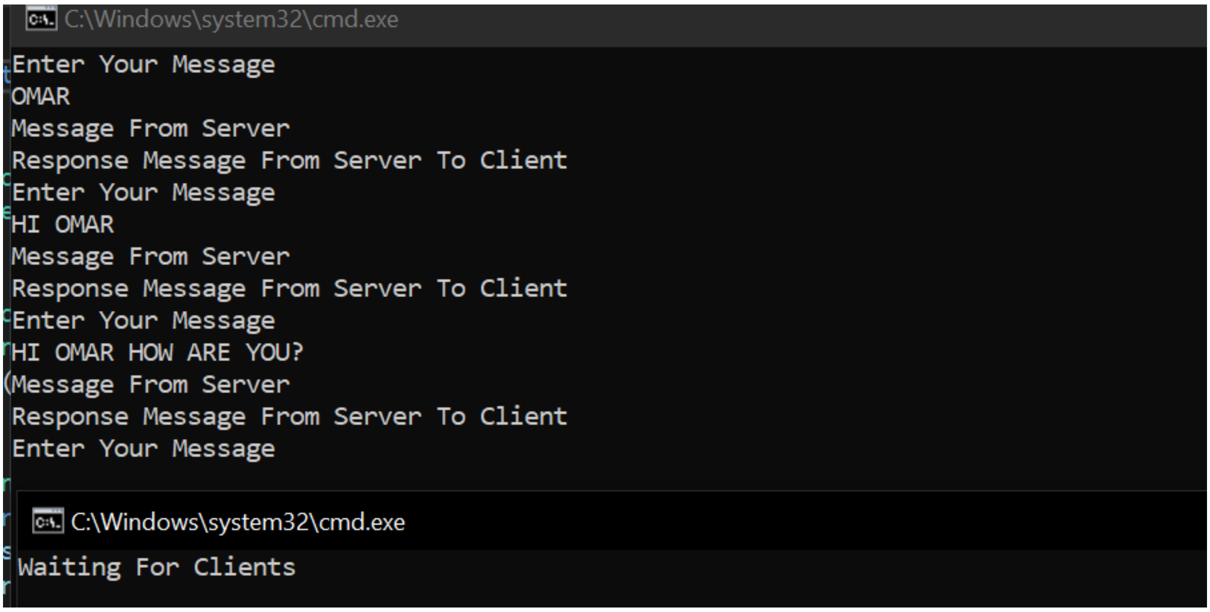
CLIENT:

```
using System;
using System.Net;
using System.Net.Sockets;
using System.Text;
class Client
{
    public static void Main()          //CLIENT
    {
        byte[] Response = new byte[1024];
        IPEndPoint EndP = new IPEndPoint(
        IPAddress.Loopback, 3000);
        Socket server = new Socket(AddressFamily.InterNetwork,
SocketType.Dgram, ProtocolType.Udp);

        IPEndPoint sender = new IPEndPoint(IPAddress.Any, 0);
        EndPoint ep = (EndPoint)sender;
        while (true)
        {

            Console.WriteLine("Enter Your Message");
```

MUHAMMAD UMAR KHAN 10619
NETWORK PROGRAMMING 108157
ASSIGNMENT#02

```
            string input = Console.ReadLine();
            Response = Encoding.ASCII.GetBytes(input);
            server.SendTo(Response, Response.Length, SocketFlags.None, EndP);
            Response = new byte[1024];
            int ToRecieve = server.ReceiveFrom(Response, ref ep);
            Console.WriteLine("Message From Server");
            Console.WriteLine(Encoding.ASCII.GetString(Response, 0,
ToRecieve));
        }
        server.Close();
    }
}
```

OUTPUT:



```
Enter Your Message
OMAR
Message From Server
Response Message From Server To Client
Enter Your Message
HI OMAR
Message From Server
Response Message From Server To Client
Enter Your Message
HI OMAR HOW ARE YOU?
Message From Server
Response Message From Server To Client
Enter Your Message
```

```
Waiting For Clients
```