

Server:

```
1  using System;
2  using System.Collections.Generic;
3  using System.Text;
4  using System.Threading.Tasks;
5  using System.Net.Sockets;
6  using System.Net;
7  using System.Threading;
8  namespace MultiChatApp_Threads_Server
9  {
10     0 references
11     class Program
12     {
13         0 references
14         static void Main(string[] args)
15         {
16             int port = 13000;
17             string IPAddress = "127.0.0.1";
18             Socket ServerListener = new Socket(AddressFamily.InterNetwork, SocketType.Stream, ProtocolType.Tcp);
19             IPEndPoint ep = new IPEndPoint(IPAddress.Parse(IPAddress), port);
20
21             ServerListener.Bind(ep);
22             ServerListener.Listen(100);
23             Console.WriteLine("Server connected ...");
24
25             Socket ServerSocket = default(Socket);
26             ServerSocket = ServerListener.Accept();
27             Console.WriteLine("Client is now connected ...");
28         }
29     }
30 }
```

```
31
32 while (true)
33 {
34     byte[] Client_Msg = new byte[50];
35     Console.WriteLine("Server receive:");
36
37     ServerSocket.Receive(Client_Msg);
38     Console.WriteLine(Encoding.ASCII.GetString(Client_Msg));
39     Console.WriteLine("Server sent:");
40
41     string str = Console.ReadLine();
42     ServerSocket.Send(Encoding.ASCII.GetBytes(str));
43 }
44 }
```

Client:

```

1  using System;
2  using System.Collections.Generic;
3  using System.Text;
4  using System.Threading;
5  using System.Threading.Tasks;
6  using System.Net;
7  using System.Net.Sockets;
8  namespace MultiChatApp_Threads_Client
9  {
10     class Program
11     {
12         static void Main(string[] args) {
13
14             Socket ClientSocket = new Socket(AddressFamily.InterNetwork,
15             SocketType.Stream, ProtocolType.Tcp);
16             IPEndPoint ep = new IPEndPoint(IPAddress.Parse(IPAddress), port);
17
18             ClientSocket.Connect(ep);
19             Program p = new Program();
20             Console.WriteLine("Connection successful ...");
21
22             Thread client_thread = new Thread(new ThreadStart(() =>
23             p.client(ClientSocket)));
24             client_thread.Start();
25         }
26
27         public void client(Socket ClientSocket)
28         {

```

```

29
30         while (true)
31         {
32             Console.WriteLine("Client sent:");
33             string str = Console.ReadLine();
34
35             ClientSocket.Send(Encoding.ASCII.GetBytes(str));
36             Console.WriteLine("Client receive:");
37             byte[] Server_Msg = new byte[1024];
38             int size = ClientSocket.Receive(Server_Msg);
39             Console.WriteLine(Encoding.ASCII.GetString(Server_Msg, 0, size));
40         }
41     }
42 }
43 }
44

```