	COLLEGE OF COMPUTING AND INFORMATION SCIENCES		
	Final-Term Assessment Spring 2021 Semester		
Class Id	106383,4,5	**Course Title**	Network programming
Program	BSCS	**Campus / Shift**	North Campus / Morning
Date	29th – April 2021	**Total Marks**	40
Duration	03 hours	**Faculty Name**	Misbah Anwer
Student Id	63361	**Student Name**	Naufil Bin Majid

Instructions:

- Filling out Student-ID and Student-Name on exam header is mandatory.
- Do not remove or change any part of exam header or question paper.
- Write down your answers in given space or at the end of exam paper with proper title "Answer for Question# _ _".
- Answers should be formatted correctly (font size, alignment and etc.)
- Handwritten text or image should be on A4 size page with clear visibility of contents.
- Only PDF format is accepted (Student are advise to install necessary software)
- In case of CHEATING, COPIED material or any unfair means would result in negative marking or ZERO.
- A mandatory recorded viva session will be conducted to ascertain the quality of answer scripts were deemed necessary.

- **Caution:** Duration to perform Final-Term Assessment is **03 hours only**. **if you failed to upload answer sheet on LMS (in PDF format) within 03 hours limit, you would be considered as ABSENT/FAILED.**
- Mention complete code with output screenshots

QUESTION NO. 1 **[5+5 Marks]**

a. Explain the following code snippet also explain text in bold from 1 to 5. Rewrite this code for helper classes.

```
IPHostEntry local = Dns.GetHostByName(Dns.GetHostName());
IPEndPoint iep = new IPEndPoint(local.AddressList[0],8000);   (1)
Socket newserver (2) = new Socket(AddressFamily.InterNetwork,
SocketType.Stream, ProtocolType.Tcp);
newserver.Bind(iep);(3)
newserver.Listen(5);(4)
Socket newclient(5) = newserver.Accept();
```

ANSWER:

In the Accept() declaration this program will interrupt, waiting for a client connection. The Socket object will provide new information about the link which should be applied for all contact with the remote client until a client communicates with the server.

The new server socket object is now connected to the original object IPEndPoint and can be used for further Accept() connections.

If Accept() no longer is called, no further client link attempts will be responded. The server will not. The client and server will start exchanging data until the client link has been approved. This function is carried out using the Receive() and Send() methods.

CLIENT:

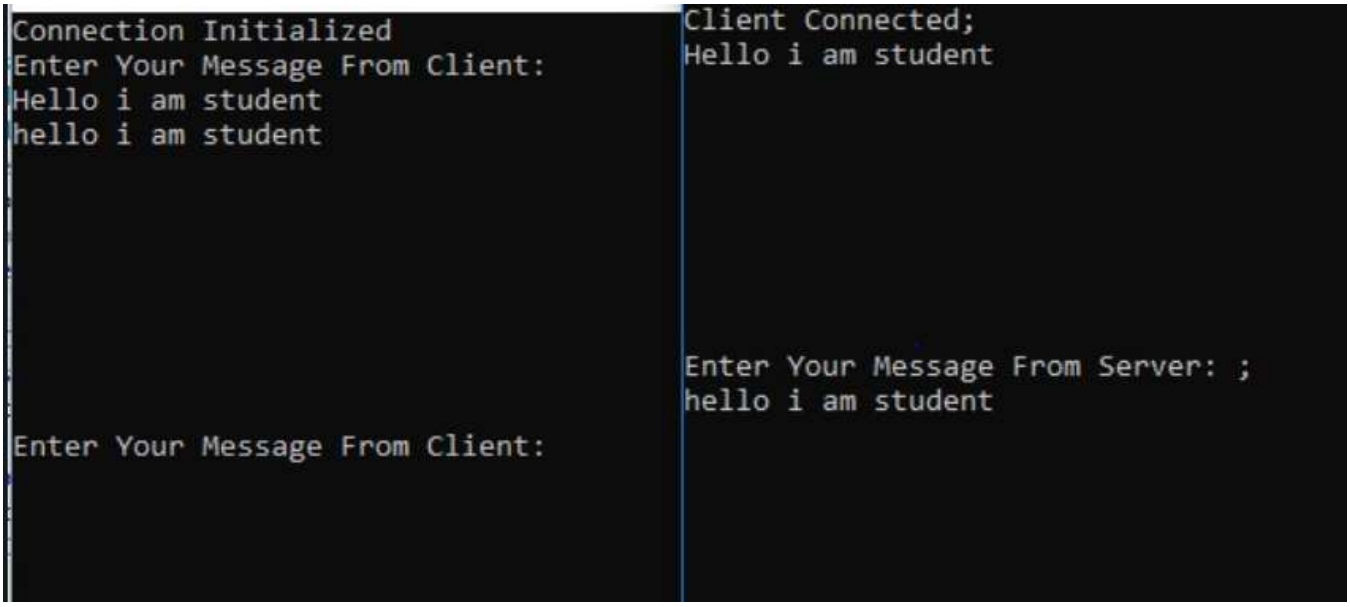
```
IPHostEntry ihe = Dns.GetHostByName(Dns.GetHostName());
        IPAddress ip = ihe.AddressList[0];
        IPEndPoint ep = new IPEndPoint(ip, 8000);
        Socket cl = new Socket(ip.AddressFamily, SocketType.Stream, ProtocolType.Tcp);
        cl.Connect(ep);
        Console.WriteLine("Connection Initialized");
        while (true)
        {
           Console.WriteLine("Enter Your Message From Client:");
           string abc =Console.ReadLine();
           cl.Send(Encoding.ASCII.GetBytes(abc));

           byte[] buffer = new byte[1024];
           cl.Receive(buffer);
           Console.WriteLine(Encoding.ASCII.GetString(buffer));
        }
```

```
        Console.Read();
```

SERVER:

```
IPHostEntry ihe = Dns.GetHostByName(Dns.GetHostName());
        IPAddress ip = ihe.AddressList[0];
        IPEndPoint ep = new IPEndPoint(ip, 8000);
        Socket sk = new Socket(ip.AddressFamily, SocketType.Stream, ProtocolType.Tcp);
        sk.Bind(ep);
        Console.WriteLine("Server Waiting;");
        sk.Listen(1);

        Socket sl = sk.Accept();
        Console.WriteLine("Client Connected;");
        while (true)
        {
           byte[] buffer = new byte[1024];

           sl.Receive(buffer);
           Console.WriteLine(Encoding.ASCII.GetString(buffer));

           Console.WriteLine("Enter Your Message From Server: ;");
           string abc = Console.ReadLine();
           sl.Send(Encoding.ASCII.GetBytes(abc));
        }
        Console.Read();
```

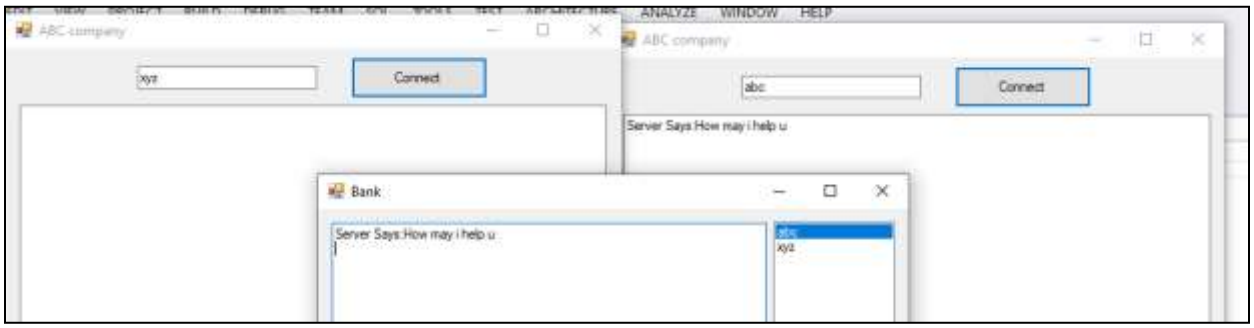
OUTPUT:



b. Suppose you are tasked to improve the throughput and reliability of real-time video feeds collection from the surveillance cameras inside the city. You incorporate this using a network of intelligent nodes with storage to collect the video feed within an appropriate geographical area. Now, ignoring how cameras storing their feeds on intelligent node, you rather focus on how to collect video feeds from the intelligent nodes in real-time to a central control room datacenter. Create an application using C# network programming for above given scenario, Make suitable assumptions.

QUESTION NO. 2 **[6+3+3 Marks]**

a. The CEO of organization need to have control on company from anywhere, the organization has three branches, branch one in Islamabad, branch two in Karachi and branch three in Lahore. You need to create asynchronous socket communication using Asynchronous methods.



ANSWER:

CLIENT:

```
namespace Client
{
    public partial class Form1 : Form
```

```
{
    public Form1()
    {
        InitializeComponent();
    }
    byte[] b = new byte[1024];
    TcpClient client = new TcpClient();

    private void textBox1_TextChanged(object sender, EventArgs e)
    {

    }

    private void textBox3_TextChanged(object sender, EventArgs e)
    {

    }

    private void button1_Click(object sender, EventArgs e)
    {
        NetworkStream ns = client.GetStream();
        StreamWriter sw = new StreamWriter(ns);
        sw.WriteLine(textBox3.Text + " Says: " + textBox2.Text);
        sw.Flush();
    }

    private void textBox2_TextChanged(object sender, EventArgs e)
    {

    }

    private void button2_Click(object sender, EventArgs e)
    {
```

```
        CheckForIllegalCrossThreadCalls = false;
        client.Connect(IPAddress.Loopback, 11000);

        NetworkStream ns = client.GetStream();
        StreamWriter sw = new StreamWriter(ns);
        sw.WriteLine("@name@" + textBox3.Text);
        sw.Flush();
        ns.BeginRead(b, 0, b.Length, ReadMsg, ns);
    }
    private void ReadMsg(IAsyncResult ar)
    {
        NetworkStream ns = (NetworkStream)ar.AsyncState;
        int count = ns.EndRead(ar);
        textBox1.Text += ASCIIEncoding.ASCII.GetString(b, 0, count);
        ns.BeginRead(b, 0, b.Length, ReadMsg, ns);

    }

    private void Form1_Load(object sender, EventArgs e)
    {

    }
  }
}
```

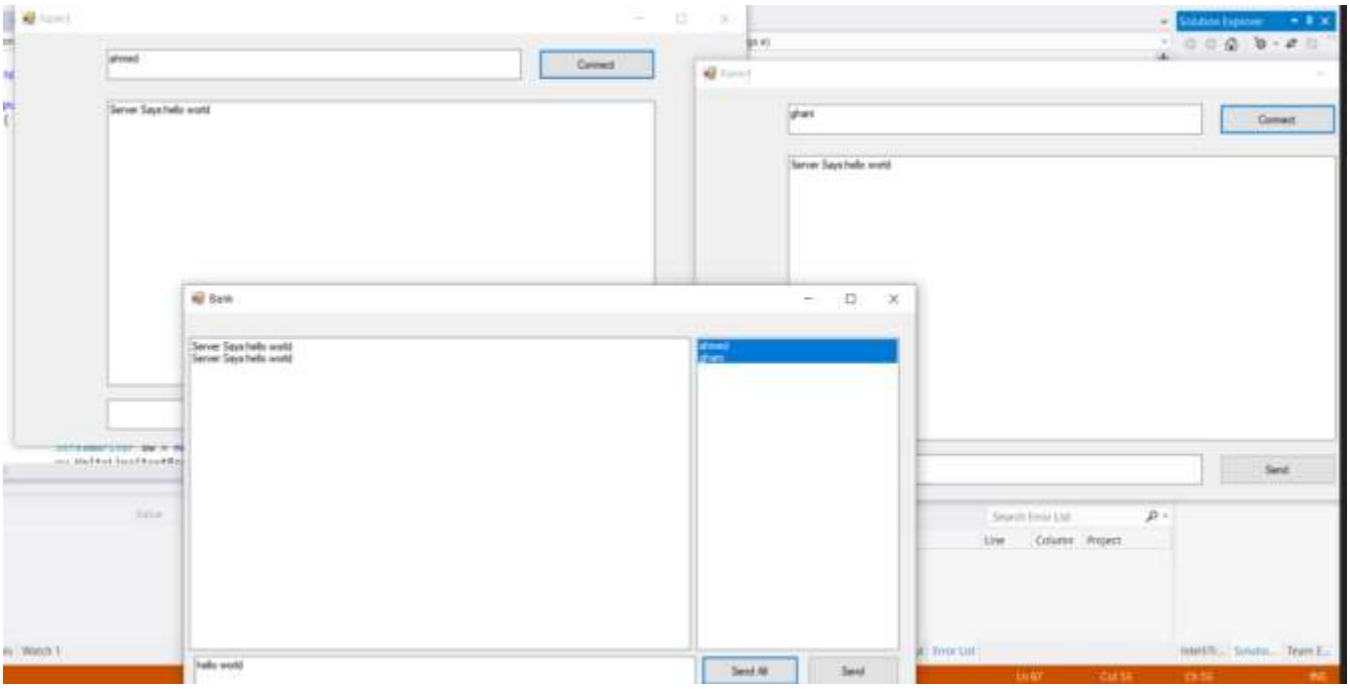
SERVER:

```
namespace Server
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e)
```

```
{
   CheckForIllegalCrossThreadCalls = false;
   TcpListener listener = new TcpListener(IPAddress.Loopback, 11000);
   listener.Start(10);
   listener.BeginAcceptTcpClient(new AsyncCallback(ClientConnect), listener);
}
Dictionary<string, TcpClient> lstClients = new Dictionary<string, TcpClient>();
byte[] b = new byte[1024];
private void ClientConnect(IAsyncResult ar)
{
   TcpListener listener = (TcpListener)ar.AsyncState;
   TcpClient client = listener.EndAcceptTcpClient(ar);
   NetworkStream ns = client.GetStream();
   object[] a = new object[2];
   a[0] = ns;
   a[1] = client;
   ns.BeginRead(b, 0, b.Length, new AsyncCallback(ReadMsg), a);
   listener.BeginAcceptTcpClient(new AsyncCallback(ClientConnect), listener);
}
private void ReadMsg(IAsyncResult ar)
{
   object[] a = (object[])ar.AsyncState;
   NetworkStream ns = (NetworkStream)a[0];
   TcpClient client = (TcpClient)a[1];
   int count = ns.EndRead(ar);
   string msg = ASCIIEncoding.ASCII.GetString(b, 0, count);
   if (msg.Contains("@name@"))
   {

      string name = msg.Replace("@name@", "");
      lstClients.Add(name, client);
      listBox1.Items.Add(name);
   }
   else
   {
      textBox1.Text += msg + Environment.NewLine;
   }
   ns.BeginRead(b, 0, b.Length, new AsyncCallback(ReadMsg), a);
}

private void button1_Click(object sender, EventArgs e)
{
   TcpClient client = (TcpClient)lstClients[listBox1.SelectedItem.ToString()];
   NetworkStream ns = client.GetStream();
   StreamWriter sw = new StreamWriter(ns);
   string textToSend = "Server Says:" + textBox2.Text;
   sw.WriteLine(textToSend);
   textBox1.Text += textToSend + Environment.NewLine;
   sw.Flush();
}

private void button2_Click(object sender, EventArgs e)
{

}
```

```
        private void textBox1_TextChanged(object sender, EventArgs e)
        {

        }
    }
}
```

OUTPUT:



b. Update part a for group communication among branches.

SERVER:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.IO;
using System.Linq;
using System.Net;
using System.Net.Sockets;
using System.Text;
using System.Windows.Forms;

namespace Server
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
```

```
}

private void Form1_Load(object sender, EventArgs e)
{
   CheckForIllegalCrossThreadCalls = false;
   TcpListener listener = new TcpListener(IPAddress.Loopback, 11000);
   listener.Start(10);
   listener.BeginAcceptTcpClient(new AsyncCallback(ClientConnect), listener);
}
Dictionary<string, TcpClient> lstClients = new Dictionary<string, TcpClient>();
byte[] b = new byte[1024];
private void ClientConnect(IAsyncResult ar)
{
   TcpListener listener = (TcpListener)ar.AsyncState;
   TcpClient client = listener.EndAcceptTcpClient(ar);
   NetworkStream ns = client.GetStream();
   object[] a = new object[2];
   a[0] = ns;
   a[1] = client;
   ns.BeginRead(b, 0, b.Length, new AsyncCallback(ReadMsg), a);
   listener.BeginAcceptTcpClient(new AsyncCallback(ClientConnect), listener);
}
private void ReadMsg(IAsyncResult ar)
{
   object[] a = (object[])ar.AsyncState;
   NetworkStream ns = (NetworkStream)a[0];
   TcpClient client = (TcpClient)a[1];
   int count = ns.EndRead(ar);
   string msg = ASCIIEncoding.ASCII.GetString(b, 0, count);
   if (msg.Contains("@name@"))
   {

      string name = msg.Replace("@name@", "");
      lstClients.Add(name, client);
      listBox1.Items.Add(name);
   }
   else
   {
      textBox1.Text += msg + Environment.NewLine;
   }
   ns.BeginRead(b, 0, b.Length, new AsyncCallback(ReadMsg), a);
}

private void button1_Click(object sender, EventArgs e)
{
   TcpClient client = (TcpClient)lstClients[listBox1.SelectedItem.ToString()];
   NetworkStream ns = client.GetStream();
   StreamWriter sw = new StreamWriter(ns);
   string textToSend = "Server Says:" + textBox2.Text;
   sw.WriteLine(textToSend);
   textBox1.Text += textToSend + Environment.NewLine;
   sw.Flush();
}

private void button2_Click(object sender, EventArgs e)
{
```

```
                //foreach (var items in listBox1.SelectedItems)
                //{
                //    TcpClient client = (TcpClient)lstClients[items.ToString()];
                //    NetworkStream ns = client.GetStream();
                //    StreamWriter sw = new StreamWriter(ns);
                //    string textToSend = "Server Says:" + textBox2.Text;
                //    sw.WriteLine(textToSend);
                //    textBox1.Text += textToSend + Environment.NewLine;
                //    sw.Flush();
                //}
            }
        }
    }
```

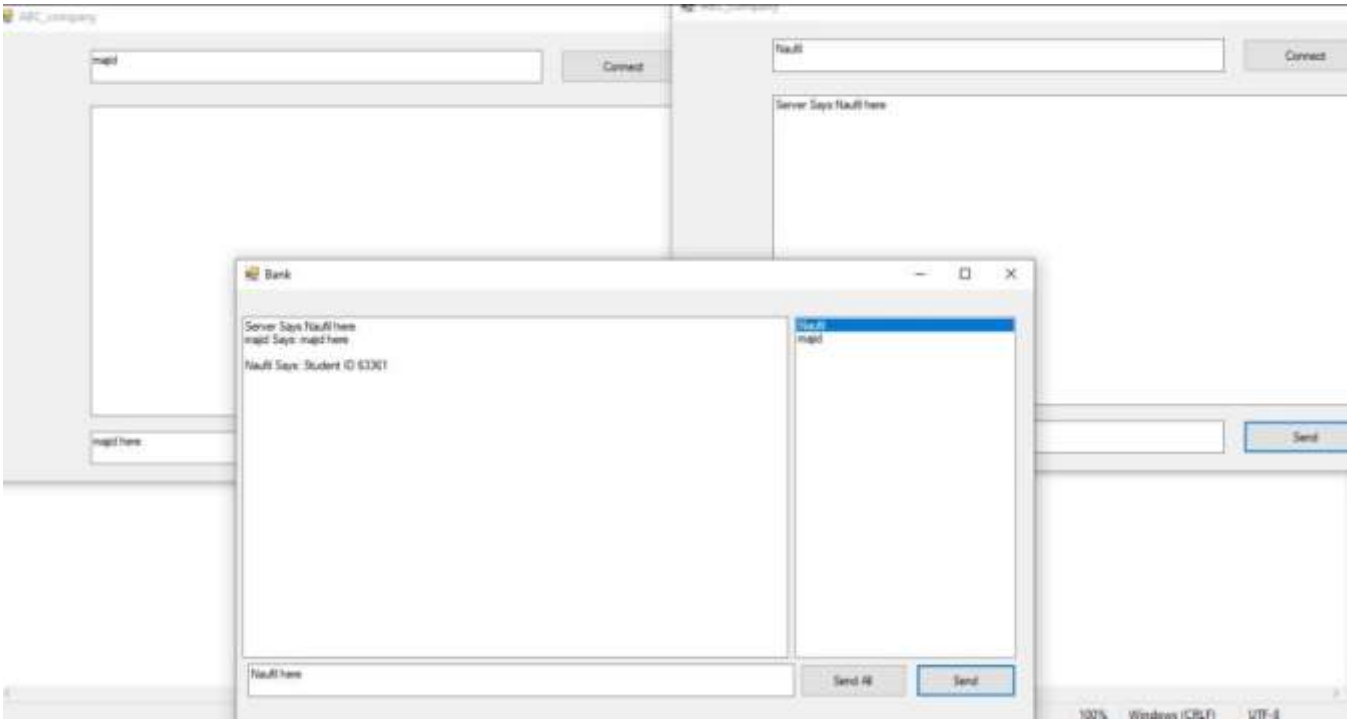
CLIENT:

```
using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.IO;

using System.Linq;

using System.Net;

using System.Net.Sockets;

using System.Text;

using System.Windows.Forms;


namespace Client

{

   public partial class Form1 : Form

   {

      public Form1()

      {

         InitializeComponent();

      }

      byte[] b = new byte[1024];

      TcpClient client = new TcpClient();


      private void textBox1_TextChanged(object sender, EventArgs e)

      {
```

```
}

private void textBox3_TextChanged(object sender, EventArgs e)
{

}

private void button1_Click(object sender, EventArgs e)
{
    NetworkStream ns = client.GetStream();
    StreamWriter sw = new StreamWriter(ns);
    sw.WriteLine(textBox3.Text + " Says: " + textBox2.Text);
    sw.Flush();
}

private void textBox2_TextChanged(object sender, EventArgs e)
{

}

private void button2_Click(object sender, EventArgs e)
{
    CheckForIllegalCrossThreadCalls = false;
    client.Connect(IPAddress.Loopback, 11000);

    NetworkStream ns = client.GetStream();
    StreamWriter sw = new StreamWriter(ns);
    sw.WriteLine("@name@" + textBox3.Text);
    sw.Flush();
    ns.BeginRead(b, 0, b.Length, ReadMsg, ns);
}
private void ReadMsg(IAsyncResult ar)
```

```
        {
            NetworkStream ns = (NetworkStream)ar.AsyncState;
            int count = ns.EndRead(ar);
            textBox1.Text += ASCIIEncoding.ASCII.GetString(b, 0, count);
            ns.BeginRead(b, 0, b.Length, ReadMsg, ns);

        }

        private void Form1_Load(object sender, EventArgs e)
        {

        }
    }
}
```

OUTPUT:



c. Define blocking, write a C# code that shows blocking also explain how to avoid blocking with the help of code

ANSWER:

Blocking is an operation that serves to prevent further performance before the other operation stops operating. TCP attachments are naturally placed in a blocking mode. This means that access is not returned to the software until those programs are not completed.

QUESTION NO. 3 **[4+4 Marks]**

a. How TCP ensure packet reliability? How we can acknowledge the UDP packets? Explain and write code. What are the parameters needed for four synchronous socket methods defined in Dns class?

ANSWER:

In contrast to UDP, TCP delivers a trustworthy message. TCP guarantees the data is not destroyed, deleted, duplicated or ordered for reception. TCP accomplish this durability by assigning each octet it transmits to a sequence number that requires a positive recognition (ACK) from the TCP that it receives.

The medium used to transmit the data is still unstable. By ensuring data meet their expected destination, TCP guarantees reliability (over unreliable medium). It does this by recognition (ACKs).If TCP collects data from the top layer (Application Layer), it divides the data into segments and attaches the header itself. This header gives this piece of data a special section no. When the receiver receives this piece, they reply to the sender by asking it to receive the next sequence of results.

There should be no explanations why the sender should not receive ACK:

1. The data had been misplaced in the middle way, and the destination thus had not received it.

2. Data have been compromised and the destination could not make sense. 2.

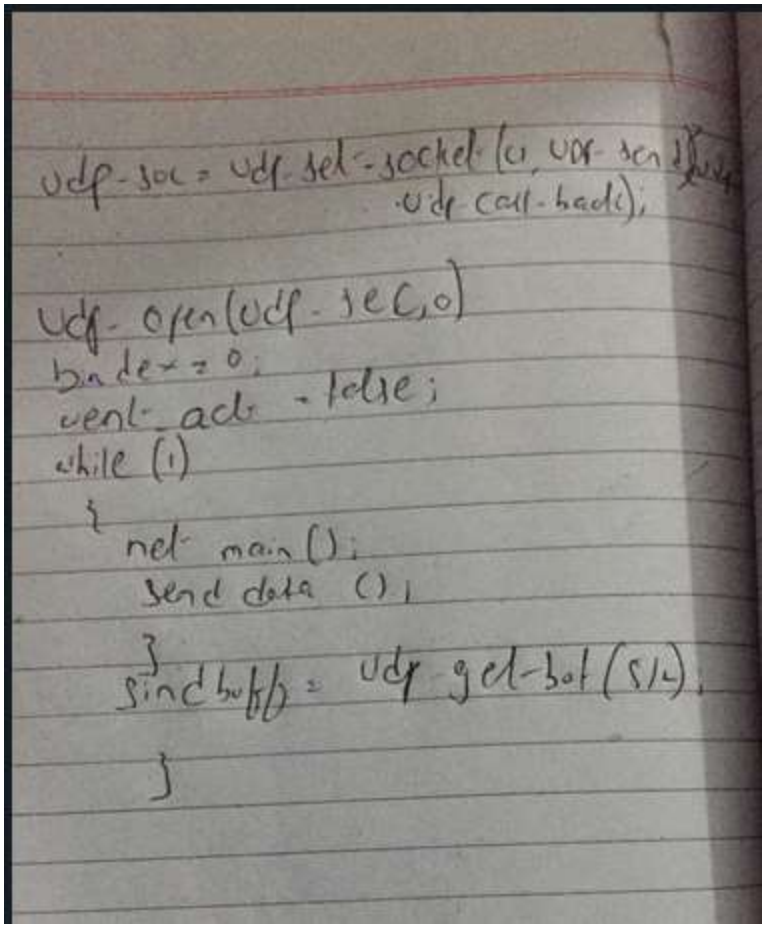
3. Missed ACK. 3.

4. Method of destination closed without sender intimation, etc.

If this ACK is not returned by the sender during the interval, send this data back and restore the timeout interval.it continues until after some no timeouts it gives up. For this function, both the sender and destination use a single random initial sequence no. The next byte of data that the other side expects. This series no.

Sender also sends an ACK to indicate to the destination for each ACK that the data is "I know. "Host (Sender/Client) does not have to send a separate ACK packet, so it will return it for the next segment chunk to be received.

On the other hand, the sender is confident that data has been obtained by the destination when ACK is received successfully.



b. Explain the issue in the following code also fix the issue (if any) and rewrite the client code

```
byte[] data = new byte[1024];
string stringData;
IPEndPoint ipep = new IPEndPoint(
IPAddress.Parse("127.0.0.1"), 9050);
Socket server = new Socket(AddressFamily.InterNetwork,
SocketType.Stream, ProtocolType.Tcp);
try
{
    server.Connect(ipep);
}
catch (SocketException e)
{
    Console.WriteLine("Unable to connect to server.");
    Console.WriteLine(e.ToString());
    return;
}
int recv = server.Receive(data);
stringData = Encoding.ASCII.GetString(data, 0, recv);
Console.WriteLine(stringData);
server.Send(Encoding.ASCII.GetBytes("message 1"));
server.Send(Encoding.ASCII.GetBytes("message 2"));
server.Send(Encoding.ASCII.GetBytes("message 3"));
server.Send(Encoding.ASCII.GetBytes("message 4"));
server.Send(Encoding.ASCII.GetBytes("message 5"));
Console.WriteLine("Disconnecting from server...");
server.Shutdown(SocketShutdown.Both);
server.Close();
```

ANSWER:

```
using System;
using System.Net;
using System.Net.Sockets;
using System.Text;
class SimpleTcpClient
{
 public static void Main()
 {
 byte[] data = new byte[1024];
 string input, stringData;
 IPEndPoint ipep = new IPEndPoint(
 IPAddress.Parse("127.0.0.1"), 9050);
 Socket server = new Socket(AddressFamily.InterNetwork,
 SocketType.Stream, ProtocolType.Tcp);
 try
 {
 server.Connect(ipep);
 } catch (SocketException e)
 {
 Console.WriteLine("Unable to connect to server.");
 Console.WriteLine(e.ToString());
 return;
 }
 int recv = server.Receive(data);
 stringData = Encoding.ASCII.GetString(data, 0, recv);
 Console.WriteLine(stringData);
 while(true)
 {
```

```
input = Console.ReadLine();
if (input == "exit")
break;
server.Send(Encoding.ASCII.GetBytes(input));
data = new byte[1024];
recv = server.Receive(data);
stringData = Encoding.ASCII.GetString(data, 0, recv);
Console.WriteLine(stringData);
}
Console.WriteLine("Disconnecting from server...");
server.Shutdown(SocketShutdown.Both);
server.Close();
}
}
```

QUESTION NO. 4 **[3+4+3 Marks]**

a. You are required to create a multimedia application for NP online Exam query session that include the following features.
   i. Instructor can upload the paper (File sharing .doc or . pdf)

ANSWER:

```
public void Send_Email() {
try
{
MailMessage mail = new MailMessage();
SmtpClient SmtpServer = new SmtpClient("smtp.gmail.com");
mail.From = new MailAddress("testmail@gmail.com");
mail.To.Add(ToUser.Text);
```

```
mail.Subject = Subject.Text;
mail.Body = Body.Text;
SmtpServer.Port = 587; //SMTP Ports: 25, 465, 587, 2525
#region
SmtpServer.Credentials = new
NetworkCredential("naufilmajid@gmail.com", "*********");
#endregion
SmtpServer.EnableSsl = true;
SmtpServer.Send(mail);
MessageBox.Show("Email Successfully Sent");
}
catch (Exception ex)
{
MessageBox.Show(ex.Message)
; }
}
public void Browse_File() {
OpenFileDialog ofd = new OpenFileDialog();
ofd.Title = "Select File";
ofd.Filter = "Image File (.png;.jpg;.bmp;.gif) |
.png;.jpg;.bmp;.gif|All files (.)|.";
ofd.ShowDialog();
if (ofd.ShowDialog() == DialogResult.OK)
{
string path = ofd.FileName.ToString();
Attach.Text = path;
}
}
```

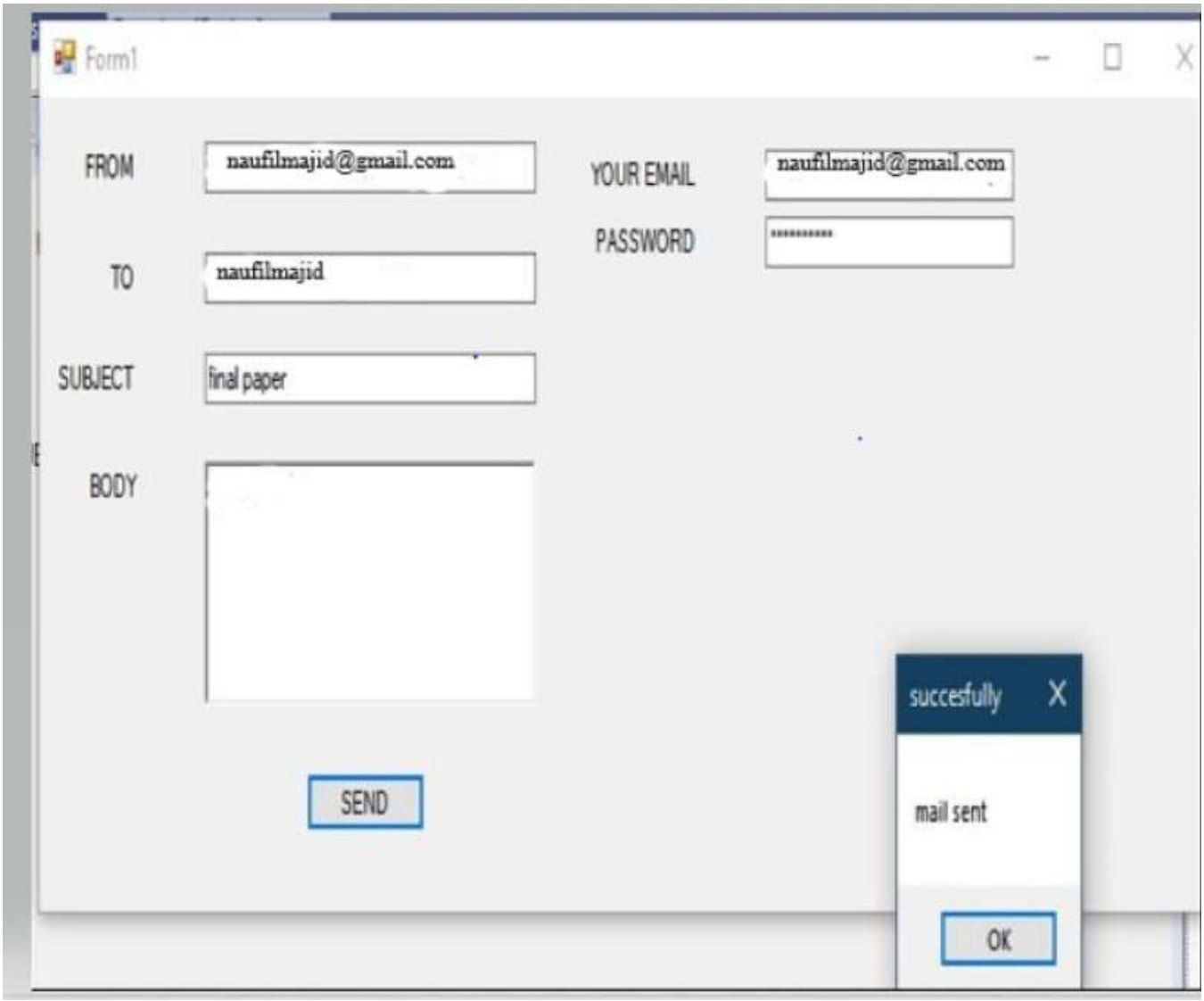
ii. Instructor can Explain the paper and Students can ask queries via chat and voice call.

b. PAF-KIET university north campus wants to transmit message to all the student about on campus examination and other updates using network functions via smtp, pop3 and IMAP. Create an application for sending the messages using mentioned protocols

ANSWER:

CODE :

```
public Form1()
        {
            InitializeComponent();
            textBox5.PasswordChar = '*';
        }

        private void button1_Click(object sender, EventArgs e)
        {
            MailMessage mail = new MailMessage(textBox1.Text, textBox2.Text, textBox3.Text,
richTextBox1.Text);
            SmtpClient client = new SmtpClient("smtp.gmail.com");
            client.Port = 587;
            client.UseDefaultCredentials = false;

            client.Credentials = new System.Net.NetworkCredential(textBox4.Text, textBox5.Text);
            client.EnableSsl = true;
            client.Send(mail);
            MessageBox.Show("mail sent", "succesfully", MessageBoxButtons.OK);
        }
```


Form1
FROM
naufilmajid@gmail.com
YOUR EMAIL
naufilmajid@gmail.com
PASSWORD
TO
naufilmajid
SUBJECT
final paper
BODY
SEND
succesfully
mail sent
OK