

Q1. Write a java program for Client Server communication using UDP Datagram Socket Programming

```
import java.io.IOException;
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;
import java.util.Scanner;

public class udpBaseClient_2
{
    public static void main(String args[]) throws IOException
    {
        Scanner sc = new Scanner(System.in);

        // Step 1: Create the socket object for
        // carrying the data.
        DatagramSocket ds = new DatagramSocket();

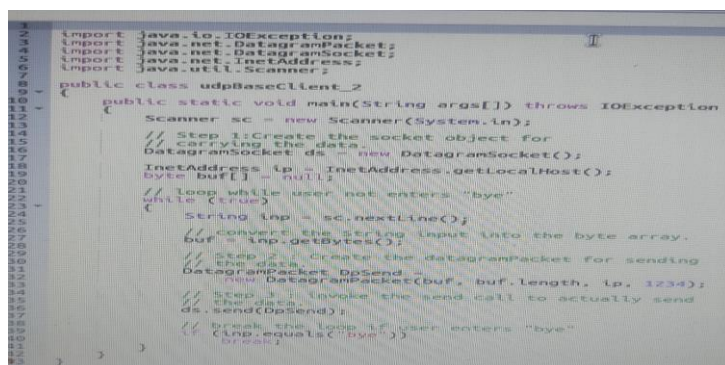
        InetAddress ip = InetAddress.getLocalHost();
        byte buf[] = null;

        // loop while user not enters "bye"
        while (true)
        {
            String inp = sc.nextLine();
            // convert the String input into the byte array.
            buf = inp.getBytes();

            // Step 2 : Create the datagramPacket for sending
            // the data.
            DatagramPacket DpSend =
                new DatagramPacket(buf, buf.length, ip, 1234);

            // Step 3 : invoke the send call to actually send
            // the data.
            ds.send(DpSend);

            // break the loop if user enters "bye"
            if (inp.equals("bye"))
                break;
        }
    }
}
```



```
1 import java.io.IOException;
2 import java.net.DatagramPacket;
3 import java.net.DatagramSocket;
4 import java.net.InetAddress;
5 import java.util.Scanner;
6
7 public class udpBaseClient_2
8 {
9     public static void main(String args[]) throws IOException
10     {
11         Scanner sc = new Scanner(System.in);
12
13         // Step 1: Create the socket object for
14         // carrying the data.
15         DatagramSocket ds = new DatagramSocket();
16
17         InetAddress ip = InetAddress.getLocalHost();
18         byte buf[] = null;
19
20         // loop while user not enters "bye"
21         while (true)
22         {
23             String inp = sc.nextLine();
24             // convert the string input into the byte array.
25             buf = inp.getBytes();
26
27             // Step 2 : Create the datagramPacket for sending
28             // the data.
29             DatagramPacket DpSend =
30                 new DatagramPacket(buf, buf.length, ip, 1234);
31
32             // Step 3 : invoke the send call to actually send
33             // the data.
34             ds.send(DpSend);
35
36             // break the loop if user enters "bye"
37             if (inp.equals("bye"))
38                 break;
39         }
40     }
41 }
```

```

import java.io.IOException;
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;
import java.net.SocketException;

public class udpBaseServer_2
{
    public static void main(String[] args) throws IOException
    {
        // Step 1 : Create a socket to listen at port 1234
        DatagramSocket ds = new DatagramSocket(1234);
        byte[] receive = new byte[65535];

        DatagramPacket DpReceive = null;
        while (true)
        {
            // Step 2 : create a DatagramPacket to receive the data.
            DpReceive = new DatagramPacket(receive, receive.length);

            // Step 3 : receive the data in byte buffer.
            ds.receive(DpReceive);

            System.out.println("Client:-" + data(receive));

            // Exit the server if the client sends "bye"
            if (data(receive).toString().equals("bye"))
            {
                System.out.println("Client sent bye.....EXITING");
                break;
            }

            // Clear the buffer after every message.
            receive = new byte[65535];
        }
    }

    // A utility method to convert the byte array
    // data into a string representation.
    public static StringBuilder data(byte[] a)
    {
        if (a == null)
            return null;
        StringBuilder ret = new StringBuilder();
        int i = 0;
        while (a[i] != 0)
        {
            ret.append((char) a[i]);
            i++;
        }
        return ret;
    }
}

```

```

    }
}

```

Q2.

Write a program to demonstrate status of key on Applet window such as KeyPressed, KeyReleased, KeyUp, KeyDown?

```

3 // Implementation using DatagramSocket
4 import java.io.IOException;
5 import java.net.DatagramPacket;
6 import java.net.DatagramSocket;
7 import java.net.InetAddress;
8 import java.net.SocketException;
9
10 public class udpBaseServer_2
11 {
12     public static void main(String[] args) throws IOException
13     {
14         // Step 1 : Create a socket to listen at port 1234
15         DatagramSocket ds = new DatagramSocket(1234);
16         byte[] receive = new byte[65535];
17
18         DatagramPacket DpReceive = null;
19         while (true)
20         {
21             // Step 2 : create a DatagramPacket to receive the data.
22             DpReceive = new DatagramPacket(receive, receive.length);
23
24             // Step 3 : receive the data in byte buffer.
25             ds.receive(DpReceive);
26
27             System.out.println("Client:-" + data(receive));
28
29             // Exit the server if the client sends "bye"
30             if (data(receive).toString().equals("bye"))
31             {
32                 System.out.println("Client sent bye.....EXITING");
33                 break;
34             }
35
36             // Clear the buffer after every message.
37             receive = new byte[65535];
38         }
39
40         // A utility method to convert the byte array
41         // data into a string representation.
42         public static StringBuilder data(byte[] a)
43         {
44             if (a == null)
45                 return null;
46             StringBuilder ret = new StringBuilder();
47             int i = 0;
48             while (a[i] != 0)
49             {
50                 ret.append((char) a[i]);
51                 i++;
52             }
53             return ret;
54         }
55     }
56 }

```

```

import
java.awt.

```

```

import
java.applet.*;

```

```

import java.awt.event.*;

```

```

public class KeyEventDemo extends Applet implements KeyListener

```

```

{
    String msg = "";

```

```

    public void init()
    {

```

```

        addKeyListener(this);
    }

```

```

    public void keyReleased(KeyEvent k)
    {

```

```

        showStatus("Key Released");
        repaint();
    }

```

```

    public void keyTyped(KeyEvent k)
    {

```

```

        showStatus("Key Typed");
        repaint();
    }

```

```

    public void keyPressed(KeyEvent k)
    {

```

```

{
    showStatus("Key Pressed");
    repaint();
}

```

```

public void paint(Graphics g)
{
    g.drawString(msg, 10, 10);
}

```

```

1  import java.awt.*;
2  import java.applet.*;
3  import java.awt.event.*;
4  public class KeyEventDemo extends Applet implements KeyListener
5  {
6      String msg = "";
7
8      public void init()
9      {
10         addKeyListener(this);
11     }
12
13     public void keyReleased(KeyEvent k)
14     {
15         showStatus("Key Released");
16         repaint();
17     }
18
19     public void keyTyped(KeyEvent k)
20     {
21         showStatus("Key Typed");
22         repaint();
23     }
24
25     public void keyPressed(KeyEvent k)
26     {
27         showStatus("Key Pressed");
28         repaint();
29     }
30
31     public void paint(Graphics g)
32     {
33         g.drawString(msg, 10, 10);
34     }
35 }
36

```

**Q3.** Write a java program to create a file with your name, save it in the desktop, write some data on the file and then read and print that data into the console

```

import java.io.*;
import java.io.IOException;
import java.util.*;

```

```

public class FileOrg {
    public static void main(String[] args) {
        try {
            File myObj = new File("filename");
            if (myObj.createNewFile())
            {
                System.out.println("File created with the name " + myObj.getName());
            }
        }
    }
}

```

```

// To read the write content on the File.....

```

```

FileWriter myWriter = new FileWriter("file name");

```

```

System.out.println("Enter Content");
Scanner input = new Scanner(System.in);
String str = input.nextLine();
myWriter.write(str);
myWriter.close();

```

```

// To show the output of the file.

```



```
System.out.println("The content of the files are as follows");
String line = null;
FileReader fileReader = new FileReader("file name");
```

```
BufferedReader bufferedReader = new BufferedReader(fileReader);
```

```
while((line = bufferedReader.readLine()) != null)
{
    System.out.println(line);
}
bufferedReader.close();
```

```
    }
    else {
        System.out.println("File already exists.");
    }
}
catch (IOException e) {
    System.out.println("An error occurred.");
    e.printStackTrace();
}
```

```
2 import java.io.*;
3 import java.io.IOException;
4 import java.util.*;
5
6 public class FileOrg {
7     public static void main(String[] args) {
8         try {
9             File myObj = new File("filename");
10            if (myObj.createNewFile())
11            {
12                System.out.println("File created with the name " + myObj.getName());
13
14                // To read the write content on the File.....
15
16                FileWriter myWriter = new FileWriter("file name");
17
18                System.out.println("Enter Content");
19                Scanner input = new Scanner(System.in);
20                String str = input.nextLine();
21                myWriter.write(str);
22                myWriter.close();
23
24                // To show the output of the file.
25
26                System.out.println("The content of the files are as follows");
27                String line = null;
28                FileReader fileReader = new FileReader("file name");
29
30                BufferedReader bufferedReader = new BufferedReader(fileReader);
31
32                while((line = bufferedReader.readLine()) != null)
33                {
34                    System.out.println(line);
35                }
36                bufferedReader.close();
37            }
38            else {
39                System.out.println("File already exists.");
40            }
41        }
42        catch (IOException e) {
43            System.out.println("An error occurred.");
44            e.printStackTrace();
45        }
46    }
47 }
```