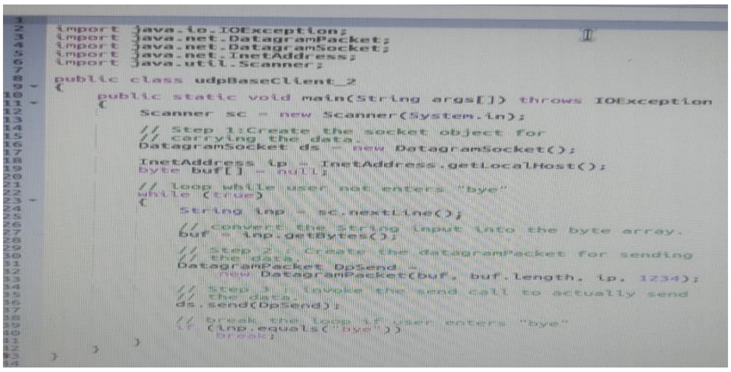
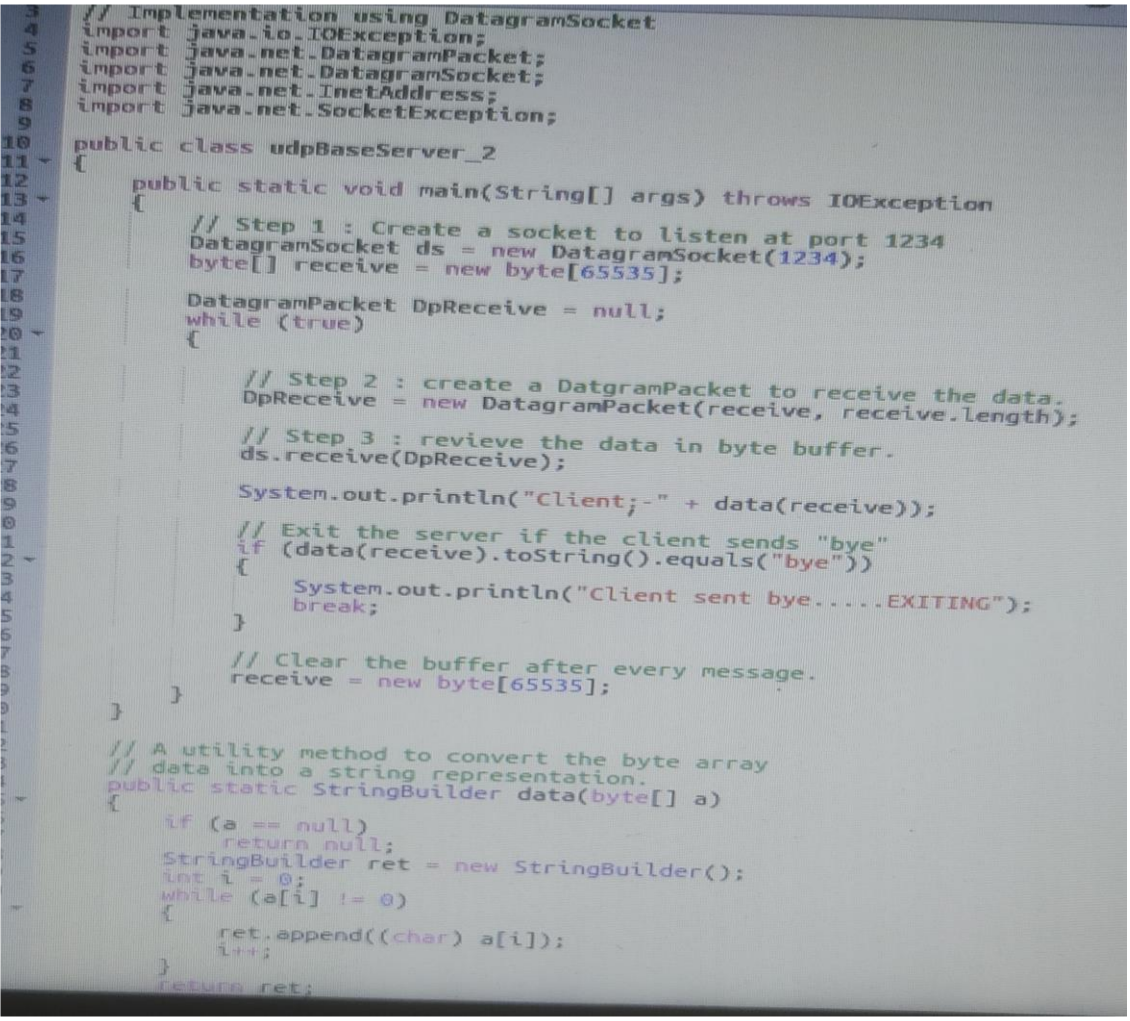
Q1. Write a java program for Client Server communication using UDP Datagram Socket Programming

```
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;
		}}
```



```
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;
        }
    }
}
```

```
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 DatgramPacket to receive the data.
                        DpReceive = new DatagramPacket(receive, receive.length);

                        // Step 3 : revieve 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;
        }
}
```

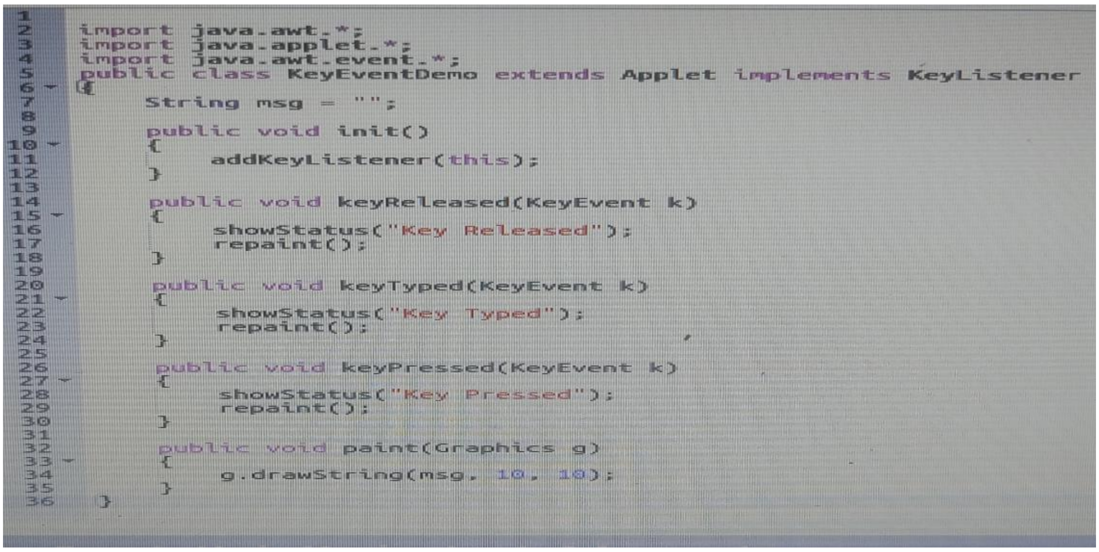


```
// Implementation using DatagramSocket
import java.io.IOException;
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;
import java.net.SocketException;

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 DatgramPacket to receive the data.
            DpReceive = new DatagramPacket(receive, receive.length);

            // Step 3 : revieve 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?

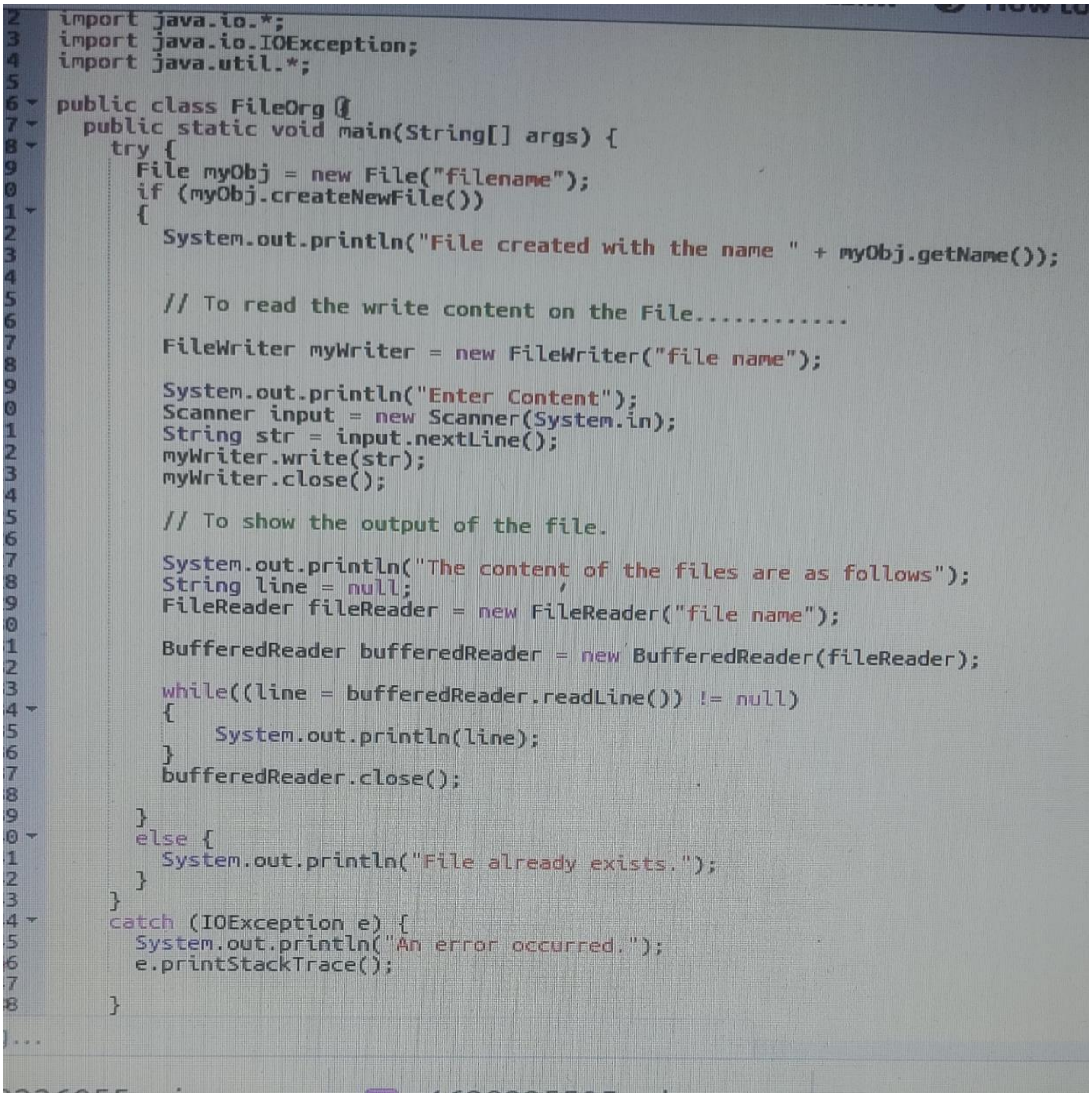
```
import java.awt.*;
import java.applet.*;
import java.awt.event.*;
public class KeyEventDemo extends Applet implements KeyListener
{
    String msg = "";

    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);
    }
}
```



```
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);
        }
}
```

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();

    }
  }
}
```



```
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();

    }
```