```java
/***************** Copyright 2002 The Hartmann Software Group, LLC   ***************/


/* Client Side Socket Program */


import java.io.*;

import java.net.*;


public class Client extends Thread
{
        private BufferedReader reader;

        private ObjectOutputStream objectout;

        private ObjectInputStream objectin;

        private Socket sock;

        private String name;

        private boolean bquit = false;


        public Client(String n) throws Exception
        {
                System.out.print("Enter in the server port or \"quit\" to end the program: ");

                reader= new BufferedReader(new InputStreamReader(System.in));

                name = n;
        }


/***************** Keyboard input information is handled in this method
*******************/
```

```java
public void KeyBoardInput() throws Exception

{

        String info = reader.readLine();

        int port = 0;


        if(info.equals("quit"))

        {

                System.exit(0);

        }

        else

                port = Integer.parseInt(info);


        sock = new Socket();


        try

        {

                sock.connect(new InetSocketAddress(InetAddress.getLocalHost(),port),2000);

        }

        catch(Exception e)

        {

                System.out.println("Unable to connect to Server");

                return;

        }


        objectout = new ObjectOutputStream(sock.getOutputStream());
```

```java
                objectout.writeObject("Name = " + name);

                start();


                while(!bquit)
                {
                        String message = reader.readLine();


                        if(message.equals("quit"))
                        {
                                objectout.writeObject(message);

                                break;
                        }
                        else
                                objectout.writeObject(message);
                }


                System.out.println("Bye");

        }


public static void main(String[] args)
{
        if(args.length == 1)
        {
                try
```

```java
            {
                    Client c = new Client(args[0]);

                    c.KeyBoardInput();

            }
            catch(Exception e)

            {
                    System.out.println("go");

                    e.printStackTrace();

            }

        }
        else

        {
            System.out.println("You must enter a client name i.e. java Client Name");

        }

    }


/****************** To maintain a socket connection, each client runs a thread
*********************/


    public void run()

    {
        try

        {
            objectin = new ObjectInputStream(sock.getInputStream());


            while(true)
```

```java
{
    Object ob = objectin.readObject();

    if(ob instanceof String)
    {
        System.out.println("Message Received: " + ob);

        if(((String)ob).equals("quit"))
        {
            objectout.writeObject("quit");
            System.out.println("The Server has unexpectedly shut down");
            System.out.println("Hit 'Enter' to terminate the program...");
            bquit = true;
            reader.close();
            break;
        }
        else if(((String)ob).equals("Bye"))
        {
            reader.close();
            System.out.println("Server connection closed");
            break;
        }
    }
    else
```

```java
                {
                        objectout.writeObject("You must send String information");
                }
        }


            System.out.println("Closed");
        }
        catch(Exception e)
        {
                e.printStackTrace();
        }
    }
}
```