**Que.1**

**Class Client:**

**import java.awt.BorderLayout;**

**import java.awt.GridLayout;**

**import java.awt.event.ActionEvent;**

**import java.awt.event.ActionListener;**

**import java.io.DataInputStream;**

**import java.io.DataOutputStream;**

**import java.io.IOException;**

**import java.net.Socket;**

**import javax.swing.JButton;**

**import javax.swing.JFrame;**

**import javax.swing.JLabel;**

**import javax.swing.JPanel;**

**import javax.swing.JScrollPane;**

**import javax.swing.JTextArea;**

**import javax.swing.JTextField;**

**/\*\***

**\***

**\* @author manthan**

**\*/**

**@SuppressWarnings("serial")**

**public class Client extends JFrame{**

**// Text field for receiving loan amount Number of years annual rate**

**private JTextField jL = new JTextField();**

**private JTextField jLayer = new JTextField();**

**private JTextField JLayerRate = new JTextField();**

**// Text area to display total rate**

**private JTextArea jTextA = new JTextArea();**

**public static JButton jbtsubmit = new JButton("submit");**

**// IO streams**

**private DataOutputStream toServer;**

**private DataInputStream fromServer;**

**public static void main(String[] args) {**

**new Client();**

**}**

**public Client() {**

**// Panel p to hold the label and text field**

**JPanel pone = new JPanel(new BorderLayout());**

**pone.setLayout(new GridLayout(3,1));**

**pone.add(new JLabel("loan amount"));**

**pone.add(new JLabel("Number of Years"));**

**pone.add(new JLabel("Annual Interest Rate"));**

**JPanel ptwo = new JPanel(new BorderLayout());**

**ptwo.setLayout(new GridLayout(3,1));**

**ptwo.add(jL);**

**ptwo.add(jLayer);**

**ptwo.add(JLayerRate);**

**JPanel pthree = new JPanel(new BorderLayout());**

**pthree.setLayout(new GridLayout(1,1));**

**pthree.add(jbtsubmit);**

**JPanel pfinal = new JPanel(new BorderLayout());**

**pfinal.setLayout(new GridLayout(1,1));**

**setLayout(new BorderLayout());**

**pfinal.add(pone);**

**pfinal.add(ptwo);**

**pfinal.add(pthree);**

**add(pfinal, BorderLayout.NORTH);**

**add(new JScrollPane(jTextA), BorderLayout.CENTER);**

**/\***

**JPanel p = new JPanel();**

**p.setLayout(new GridLayout(4,2));**

**p.add(new JLabel("loan amount"));**

**p.add(jL);**

**p.add(new JLabel("Number of Years"));**

**p.add(jLayer);**

**p.add(new JLabel("Annual Interest Rate"));**

**p.add(JLayerRate);**

**p.add(jbtsubmit);**

**setLayout(new BorderLayout());**

**add(p, BorderLayout.NORTH);**

**add(new JScrollPane(jTextA), BorderLayout.CENTER);**

**\*/**

**setTitle("Client");**

**setSize(500, 300);**

**setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);**

**setVisible(true); // It is necessary to show the frame here!**

**try {**

**// Create a socket to connect to the server**

**@SuppressWarnings("resource")**

**Socket socket = new Socket("localhost", 8000);**

**fromServer = new DataInputStream(socket.getInputStream());**

**toServer = new DataOutputStream(socket.getOutputStream());**

**}**

**catch (IOException ex) {**

**jTextA.append(ex.toString() + '\n');**

**}**

**jbtsubmit.addActionListener(new ActionListener() {**

**@Override**

**public void actionPerformed(ActionEvent ae) {**

**try{**

**jTextA.append("Loan amount"+jL.getText()+ "\n");**

**jTextA.append("Total number of year"+jLayer.getText()+ "\n");**

**jTextA.append("Annual Interest Rate"+JLayerRate.getText()+ "\n");**

**@SuppressWarnings("unused")**

**Loan l = new Loan(Float.parseFloat(JLayerRate.getText().trim()),Integer.parseInt(jLayer.getText().trim()),Double.parseDouble((jL.getText().trim())));**

**toServer.writeDouble(Double.parseDouble(jL.getText().trim()));**

**toServer.writeInt(Integer.parseInt(jLayer.getText().trim()));**

**toServer.writeFloat(Float.parseFloat(JLayerRate.getText().trim()));**

**jTextA.append("Monthly payment" + String.valueOf(fromServer.readDouble()) + "\n");**

**jTextA.append("Total payment" +String.valueOf(fromServer.readInt())+ "\n");**

**}**

**catch(Exception e)**

**{**

**System.out.println(e);**

**}**

**}**

**});**

**}**

**}**

Class Loan:

**public** **class** Loan {

**private** **float** anIntR;

**private** **int** NumYear;

**private** **double** lAmount;

**private** java.util.Date lDate;

/\*\* Default constructor \*/

**public** Loan() {

**this**.anIntR = (**float**)2.5;

**this**.NumYear = 1;

**this**.lAmount =1000;

}

/\*\* Construct a loan with specified annual interest rate,

number of years, and loan amount

\*/

**public** Loan(**float** anIntR, **int** NumYear,

**double** lAmount) {

**this**.anIntR = anIntR;

**this**.NumYear = NumYear;

**this**.lAmount = lAmount;

lDate = **new** java.util.Date();

}

/\*\* Return anIntR \*/

**public** **double** getanIntR() {

**return** anIntR;

}

/\*\* Set a new anIntR \*/

**public** **void** setanIntR(**float** anIntR) {

**this**.anIntR = anIntR;

}

/\*\* Return NumYear \*/

**public** **int** getNumYear() {

**return** NumYear;

}

/\*\* Set a new NumYear \*/

**public** **void** setNumYear(**int** NumYear) {

**this**.NumYear = NumYear;

}

/\*\* Return lAmount \*/

**public** **double** getlAmount() {

**return** lAmount;

}

/\*\* Set a newlAmount \*/

**public** **void** setlAmount(**double** lAmount) {

**this**.lAmount = lAmount;

}

/\*\* Find monthly payment \*/

**public** **double** getMonthlyPayment() {

**double** monthlyInterestRate = anIntR / 1200;

**double** monthlyPayment = lAmount \* monthlyInterestRate / (1 -

(1 / Math.*pow*(1 + monthlyInterestRate, NumYear \* 12)));

**return** monthlyPayment;

}

/\*\* Find total payment \*/

**public** **double** getTotalPayment() {

**double** totalPayment = getMonthlyPayment() \* NumYear \* 12;

**return** totalPayment;

}

/\*\* Return loan date \*/

**public** java.util.Date getlDate() {

**return** lDate;

}

}

Class Server:

**import** java.io.\*;

**import** java.net.\*;

**import** java.util.\*;

**import** java.awt.\*;

**import** javax.swing.\*;

@SuppressWarnings("serial")

**public** **class** Server **extends** JFrame {

// Text area for displaying contents

**private** JTextArea jTextA = **new** JTextArea();

**public** **static** **void** main(String[] args) {

**new** Server();

}

**public** Server() {

// Place text area on the frame

setLayout(**new** BorderLayout());

add(**new** JScrollPane(jTextA), BorderLayout.***CENTER***);

setTitle("Server");

setSize(500, 300);

setDefaultCloseOperation(JFrame.***EXIT\_ON\_CLOSE***);

setVisible(**true**); // It is necessary to show the frame here!

**try** {

// Create a server socket

@SuppressWarnings("resource")

ServerSocket serverSocket = **new** ServerSocket(8000);

jTextA.append("Server started at " + **new** Date() + '\n');

// Listen for a connection request

Socket socket = serverSocket.accept();

// Create data input and output streams

DataInputStream inputFromClient = **new** DataInputStream(socket.getInputStream());

DataOutputStream outputToClient = **new** DataOutputStream(socket.getOutputStream());

jTextA.append("server working\n");

**while** (**true**) {

jTextA.append("Server inside from while \n");

**double** loanamount = inputFromClient.readDouble();

**int** NumberOfYears = inputFromClient.readInt();

**float** totalInterestRate = inputFromClient.readFloat();

// Compute area

Loan l = **new** Loan(totalInterestRate, NumberOfYears, loanamount);

**double** monthlyPayment = l.getMonthlyPayment();

**double** totalPayment = l.getTotalPayment();

// Send area back to the client

outputToClient.writeDouble(monthlyPayment);

outputToClient.writeDouble(totalPayment);

jTextA.append("Loan amount received from client: " +String.*valueOf*(loanamount)+'\n');

jTextA.append("Interest Rate received from client: " +String.*valueOf*(totalInterestRate) + '\n');

jTextA.append("Total Number of years reserived from client: " + String.*valueOf*(NumberOfYears) + '\n');

jTextA.append("Monthly Payment : " + String.*valueOf*(monthlyPayment) + '\n');

jTextA.append("Total Payment : " + String.*valueOf*(totalPayment) + '\n');

}

}

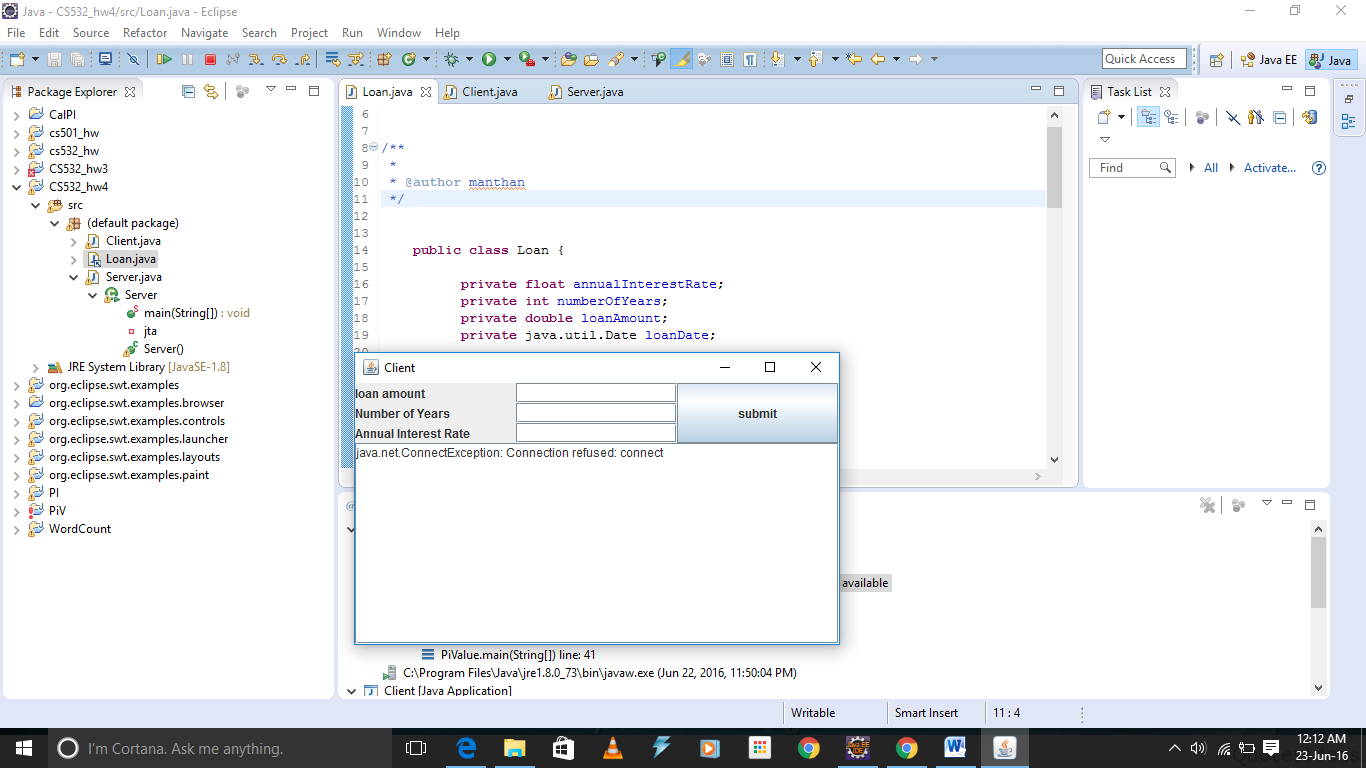
**catch**(IOException ex) {

System.***err***.println(ex);

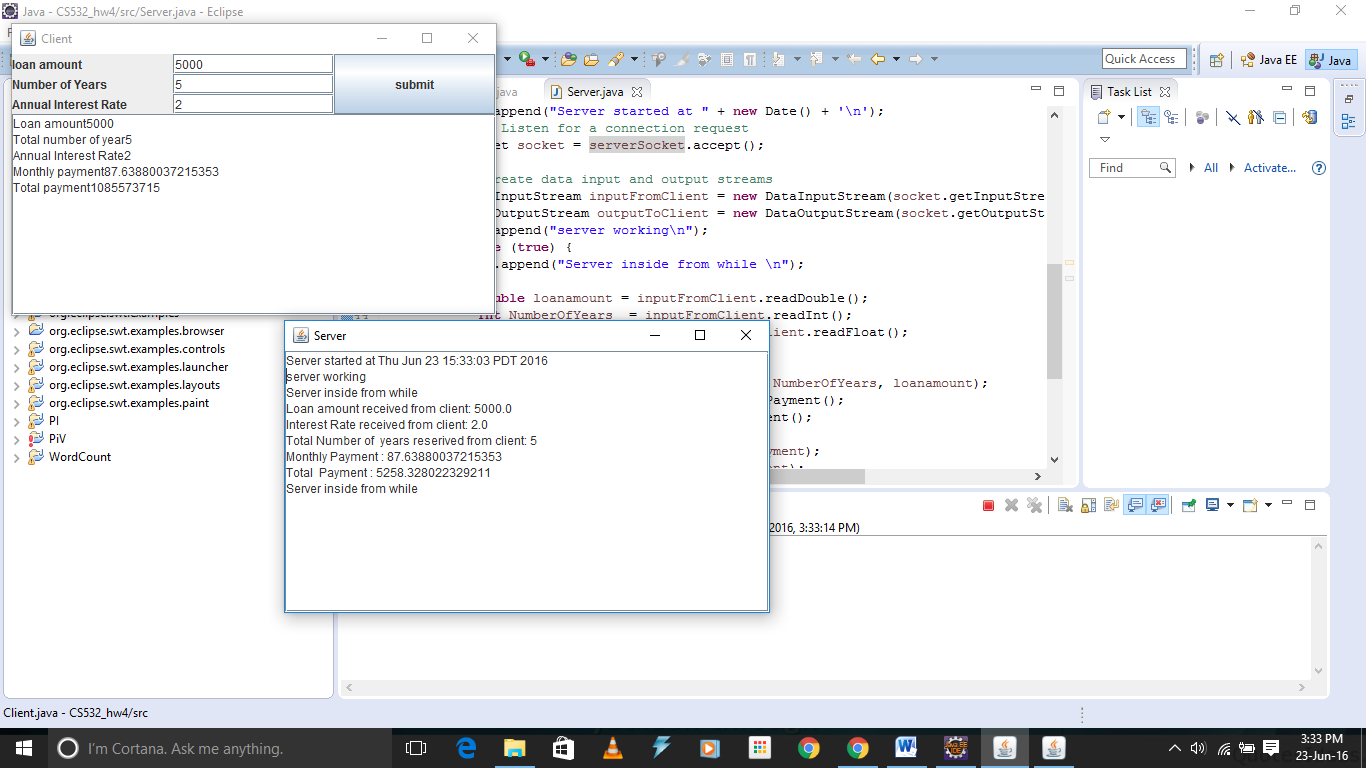
}

}

}



**Total Details:**



**Que 2:**

**import java.awt.\*;**

**import java.awt.event.\*;**

**import java.io.\*;**

**import java.net.\*;**

**import javax.swing.\*;**

**public class ChatClient extends JFrame implements Runnable**

**{**

**private static final long serialVersionUID = 1L;**

**private JTextField name = new JTextField("Enter a name");**

**private JTextField text = new JTextField();**

**private JTextArea clientTextArea = new JTextArea();**

**private Socket socket;**

**private DataOutputStream dataout;**

**private DataInputStream datain;**

**//main**

**public static void main(String[] args) {**

**new ChatClient();**

**}**

**//Constructor**

**public ChatClient() {**

**JPanel panel1 = new JPanel();**

**panel1.setLayout(new BorderLayout());**

**panel1.add(new JLabel("Enter text"), BorderLayout.WEST);**

**panel1.add(text, BorderLayout.CENTER);**

**JPanel panel2 = new JPanel();**

**panel2.setLayout(new BorderLayout());**

**panel2.add(new JLabel("Name"), BorderLayout.WEST);**

**panel2.add(name, BorderLayout.CENTER);**

**JPanel panel = new JPanel();**

**panel.setLayout(new BorderLayout());**

**panel.add(panel1, BorderLayout.SOUTH);**

**panel.add(panel2, BorderLayout.NORTH);**

**setLayout(new BorderLayout());**

**add(panel, BorderLayout.NORTH);**

**add(new JScrollPane(clientTextArea), BorderLayout.CENTER);**

**clientTextArea.setEditable(false);**

**text.addActionListener(new ButtonClickListener());**

**setTitle("Multi Chat Client");**

**setSize(500,300);**

**setLocationRelativeTo(null);**

**setLocation(1200,400);**

**setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);**

**setVisible(true);**

**try {**

**socket = new Socket("localhost", 8000);**

**datain = new DataInputStream(socket.getInputStream());**

**dataout = new DataOutputStream(socket.getOutputStream());**

**new Thread(this).start();**

**}catch(IOException ex) {**

**clientTextArea.append(ex.toString() + "\n");**

**}**

**}**

**private class ButtonClickListener implements ActionListener {**

**@Override**

**public void actionPerformed(ActionEvent e) {**

**try{**

**String data = name.getText().trim() + ": " + text.getText().trim(); //Get the text from the text field**

**dataout.writeUTF(data); // Send the text to the server**

**text.setText("");**

**}catch (IOException ex){**

**System.err.println(ex);**

**}**

**}**

**}**

**public void run(){**

**try{**

**while(true){**

**String text1 = datain.readUTF(); // Get message**

**clientTextArea.append(text1 + "\n"); //Display to the text area**

**}**

**}catch(IOException ex){**

**System.err.println(ex);**

**}**

**}**

**}**

**import** java.awt.\*;

**import** java.io.\*;

**import** java.net.\*;

**import** java.util.\*;

**import** javax.swing.\*;

**public** **class** ChatServer **extends** JFrame{

**private** **static** **final** **long** ***serialVersionUID*** = 1L;

//Text area for displaying contents

**private** JTextArea serverTextArea = **new** JTextArea();

**private** Hashtable<Socket, DataOutputStream> Streams = **new** Hashtable<Socket, DataOutputStream>();

**private** ServerSocket serverSocket; //Server socket

//main

**public** **static** **void** main(String[] args) {

**new** ChatServer();

}

//Constructor

**public** ChatServer() {

//Place text area on the frame

setLayout(**new** BorderLayout());

add(**new** JScrollPane(serverTextArea), BorderLayout.***CENTER***);

setTitle("Multi Chat Server");

setSize(650, 300);

setDefaultCloseOperation(JFrame.***EXIT\_ON\_CLOSE***);

setVisible(**true**);

setLocation(1150,100);

serverTextArea.setEditable(**false**);

listen();

}

**private** **void** listen() {

**try** {

serverSocket = **new** ServerSocket(8000); // Create a server socket

serverTextArea.append("Multi Thread Server started at " + **new** Date() + '\n');

**while** (**true**){

Socket socket = serverSocket.accept(); //Listen for a new connection request

serverTextArea.append("Connection from " + socket + " at " + **new** Date() + '\n');

DataOutputStream dataout = **new** DataOutputStream(socket.getOutputStream());

Streams.put(socket, dataout);

**new** ServerThread(**this**, socket);

}

}

**catch**(IOException ex) {

System.***err***.println(ex);

}

}

//Used to get the output streams

Enumeration<DataOutputStream> getOutputStreams(){

**return** Streams.elements();

}

//Used to send message to all clients

**void** sendToAll(String message){

//Go through hashtable and send message to each output stream

**for**(Enumeration<?> e = getOutputStreams(); e.hasMoreElements();)

{

DataOutputStream dout = (DataOutputStream)e.nextElement();

**try**{

dout.writeUTF(message); // Write message

}**catch** (IOException ex){

System.***err***.println(ex);

}

}

}

**class** ServerThread **extends** Thread {

**private** ChatServer server;

**private** Socket socket;

**public** ServerThread(ChatServer server, Socket socket) {

**this**.socket = socket;

**this**.server = server;

start();

}

**public** **void** run() {

**try** {

DataInputStream inputStream = **new** DataInputStream(socket.getInputStream());

**while** (**true**) {

String data = inputStream.readUTF();

server.sendToAll(data);

serverTextArea.append(data +"\n");

}

}**catch**(IOException e) {

System.***err***.println(e);

}

}

}

}

