Ex.No.1 21/Jan/2020	SOCKET PROGRAMMING – TCP & UDP

AIM:

To Write programs for

- Finding gst for 3 different products

using Socket Programming.

THEORY:

Socket:

A network socket is an internal endpoint for sending and receiving data within a node on a computer network. A socket is a software object that acts as an endpoint establishing a bidirectional network communication link between a server side and a client-side program.

Differences between TCP and UDP:

S.No	TCP	UDP
1	TCP stands for Transmission Control Protocol	UDP stands for User Datagram Protocol
2	It is Connection Oriented Protocol	It is a Connection less protocol.
3	TCP establishes a connection between the computers before transmitting the data.	UDP transmits the data directly to the destination without checking whether the system is ready to receive or not
4	Slow transmission – low speed	Fast transmission-high Speed
5	Highly reliable	Un reliable

Registered Ports and Examples (Any seven):

The range of port numbers from 1024 to 49151 are the registered ports. They are assigned by IANA for specific service upon application by a requesting entity. On most systems, registered ports can be used without superuser privileges.

1.1167 – Cisco IP SLA(Service Assurance Agent)

2.1029 - Microsoft DCOM services

3.1059 - nimreg, IBM AIX Network Installation Manager (NIM)

4.1080 – SOCKS proxy

5.1085 – WebObjects

6.1098 – rmiactivation, Java remote method invocation(RMI) activation

7.1099 – rmiregistry, Java remote method invocation(RMI) registry

IANA:

Internet Assigned Numbers Authority is a function of ICANN, a nonprofit American corporation that oversees a global IP address allocation, autonomous system number allocation, root zone management in Domain Name System (DNS), media types and other Internet Protocol related symbols and internet numbers.

Process for TCP Socket connection establishment:

1. Sender starts the process.
2. TCP is a full duplex protocol so both sender and receiver require a window for receiving messages from one another.
3. Sender makes the final reply for connection establishment.

Client side

- Create a socket using the socket() function;
- Connect the socket to the address of the server using the connect() function;
- Send and receive data by means of the read() and write() functions.

Server Side

- Create a socket with the socket() function;
- Bind the socket to an address using the bind() function;
- Listen for connections with the listen() function;
- Accept a connection with the accept() function system call. This call typically blocks until a client connects with the server.

Process for UDP Socket connection establishment:

Client Side

- Create a socket using the socket() function;
- Send and receive data by means of the recvfrom() and sendto() functions.

Server Side

- Create a socket with the socket() function;
- Bind the socket to an address using the bind() function;
- Send and receive data by means of recvfrom() and sendto().

ALGORITHM:

TCP

Client:

1. Start
2. Import required java packages.
3. In Tclient class , create the object with IP address and port number for the class Socket.
4. Create object for PrintWriter with socket.getOutputStream() method.
5. Create object for BufferedReader with InputStreamReader and socket.getInputStream().
6. Create object for Scanner class for getting input from the users.
7. Display the list of products with choices and get choice and product from users.
8. Send the choice and product name to the Server with PrintWriter object .
9. If choice equals to 1 then get the reply from the server by readLine() method with BufferedReader object and print the gst for product 1.
10. If choice equals to 2 then get the product 2 from server readLine() method with BufferedReader object and displayed.
11. If choice equals to 3 then read the starting index from user , send the index to the server and display gst for product 3.
12. If choice equals to 4 then get the reply from server and end the process.
13. Close the connection.
14. End

Server

1. Start
2. Import required java packages.
3. In TServer class,create object for the class ServerSocket with port number.
4. Accept the request from client by creating object for Socket and call the accept()method.
5. Create object for PrintWriter with socket.getOutputStream() method.

6. Create object for BufferedReader with InputStreamReader and socket.getInputStream().
7. Receive the string and choice from client with BufferedReader object and readLine() method.
8. If choice equals to 1 then find the gst for product 1 using user defined function and convert it into integer and send to the client.
9. If choice equals to 2 then product 2 and find the gst for the product 2 and send to the client.
10. If choice equals to 3 then product 3 and find the gst for the product 3 and send to the client.
11. If choice equals to 4 then display the total cost and end the connection
12. Close the connection
13. End

CODING:

Client

```
import java.net.*;
import java.io.*;
import java.util.*;

public class Client
{
private Socket socket
= null;
private DataInputStream input = null;
private DataOutputStream out                                                    = null;
public Client(String address, int port)
{
Scanner sc=new Scanner(System.in);
try
{
socket = new Socket(address, port);
System.out.println("Connected");
input = new DataInputStream(System.in);
out = new DataOutputStream(socket.getOutputStream());
}
catch(UnknownHostException u)
{
System.out.println(u);
}
catch(IOException i)
{
System.out.println(i);
}
int a,n,money,l;
String line = "";
while (!line.equals("Over"))
{
try
{

//line=input.readUTF();
line=sc.nextLine();
out.writeUTF(line);
}
catch(IOException i)
{
System.out.println(i);
}
}

// close the connection
try
{
input.close();
out.close();
socket.close();
}
catch(IOException i)
{
System.out.println(i);
}
```

```
}

public static void main(String args[])
{
Client client = new Client("127.0.0.1", 4000);
}
}
```

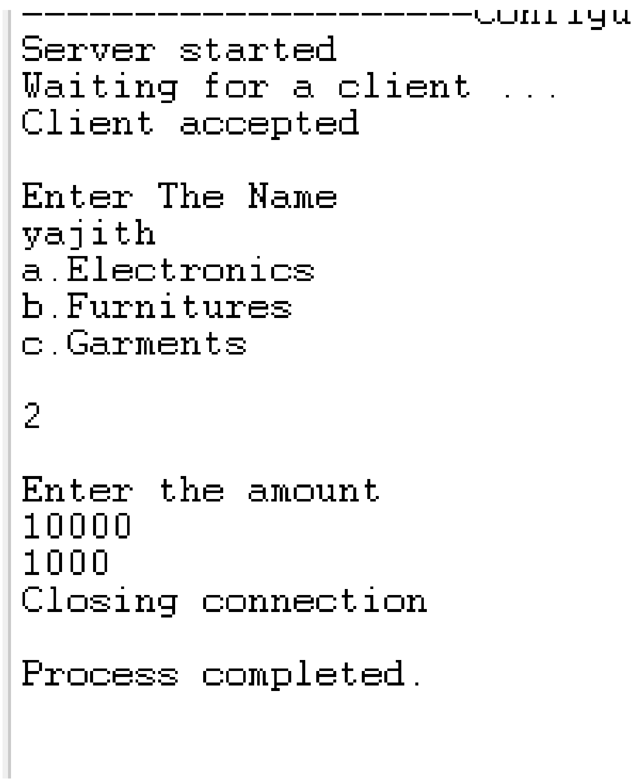
Server

```
import java.net.*;
import java.io.*;

public class Server
{
private Socket
socket = null;
private ServerSocket server = null;
private DataInputStream in = null;
public static int gst(int l,int money)
{
int gstamount=(money*l)/100;
return gstamount;
}
public Server(int port)
{
int a,n,money,l;
try
{
server = new ServerSocket(port);
System.out.println("Server started");

System.out.println("Waiting for a client ...");
socket = server.accept();
System.out.println("Client accepted");

// takes input from the client socket
```

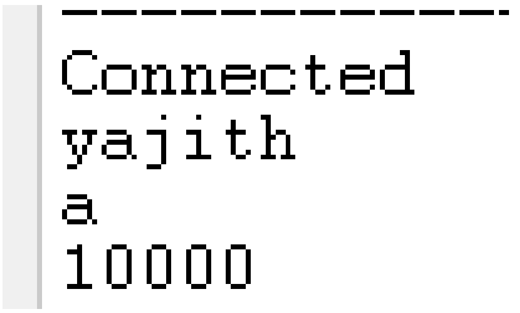
```
in = new DataInputStream(
new BufferedInputStream(socket.getInputStream()));

String line = "";

// reads message from client until "Over" is sent
while (!line.equals("Over"))
{
try
{
String list[]=new String[]{"a.Electronics","b.Furnitures","c.Garments"};
int[] gstper=new int[]{20,13,5};
String name="";
System.out.println("\nEnter The Name");
line=in.readUTF();
System.out.println(line);
name=line;
for(int i=0;i<3;i++)
{
System.out.println(list[i]);
}
line=in.readUTF();
System.out.println(line);
if(line.equals("a"))
{
n=0;
}
else if(line.equals('b'))
{
n=1;
}
else
{
n=2;
}
System.out.println(n);
System.out.println("\nEnter the amount");
line=in.readUTF();
money=10000;
System.out.println(money);
l=10;

if(n==0)
{
a=gst(l,money);
System.out.println(a);
line="Over";
}
if(n==1)
{
a=gst(l,money);
System.out.println(a);
line="Over";
}
if(n==2)
{
a=gst(l,money);
System.out.println(a);
line="Over";
}

}
catch(IOException i)
{
System.out.println(i);
}
}
System.out.println("Closing connection");
socket.close();
in.close();
}
catch(IOException i)
{
System.out.println(i);
```

```
}
}

public static void main(String args[])
{
Server server = new Server(4000);
}
}
```

SCREEN SHOTS:



```
Server started
Waiting for a client ...
Client accepted

Enter The Name
yajith
a.Electronics
b.Furnitures
c.Garments

2

Enter the amount
10000
1000
Closing connection

Process completed.
```



```
Connected
yajith
a
10000
```

ALGORITHM:

UDP

Client

1. Start
2. Import necessary java packages
3. In Clientu class Create objects for InetAddress,DatagramSocket and DatagramPacket.
4. Create byte array to send and receive the messages.
5. Get IP address with InetAddress.getByName()method.
6. Initialise DatagramSocket object .
7. Get the choice and string from user.
8. Convert user inputs to byte array by input.getBytes() method.
9. Send Choice and string to the server by the DatagramPacket object, DatagramSocket object with send(array,array length,IP address,port number) method.
10. If choice equals to one then
    i. Initialise DatagramPacket object for receiving packet.
    ii. It receives the packet from server by DatagramPacket and DatagramSocket object with object.receive(receiving object).
    iii. Get the Name of product and total amount with the help of object.getData() method.
    iv. Convert the gst and display it
11. If choice equals to two then ,
    i. Initialise DatagramPacket object for receiving packet.
    ii. It receives the packet from server by DatagramPacket and DatagramSocket object with object.receive(receiving object).
    iii. Get the product name and amount with the help of object.getData() method and display it.
12. If choice equals to three then,
    i. Get the product name and total amount of user.
    ii. Convert it into byte array and send to the server.
13. If choice equals to four then display the gst and total amount to be paid.
14. End

Server

```
1. Start
2. Import necessary java packages
3.In Serveru class Create objects for InetAddress,DatagramSocket and DatagramPacket.
4. Create byte array to send and receive the messages.
5. Initialise DatagramSocket object with Port number.
6. Receive the choice and string from client using receivepacket and receive () method.
7. If choice equals to one then
    i.   find gst of the product with string.length() method
   ii.   Convert  it into string and then byte array
  iii.   Send to the client by DatagramSocket and DatagramPacket object with send()
         method.
8. If choice equals to two then Convert it into byte array and send to the client.
9.If choice equals to three then,
    i.   Receive the starting and end index.
   ii.   Convert it into gst for the product
  iii.   Convert the substring into byte array and send to the client.
10. If choice equals to four then total amount is displayed and process ends
11.Close Connection
12.End
```

CODING:

Client

```
// Java program to illustrate Client side
// Implementation using DatagramSocket
import java.io.IOException;
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;
import java.util.Scanner;

public class ClientEcho
{
```

```
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;
System.out.println("Enter the name\nEnter the amount");
System.out.print("1.Electronics,2.Garments,3.Furnitures\n");

// 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;
}
}
}
```

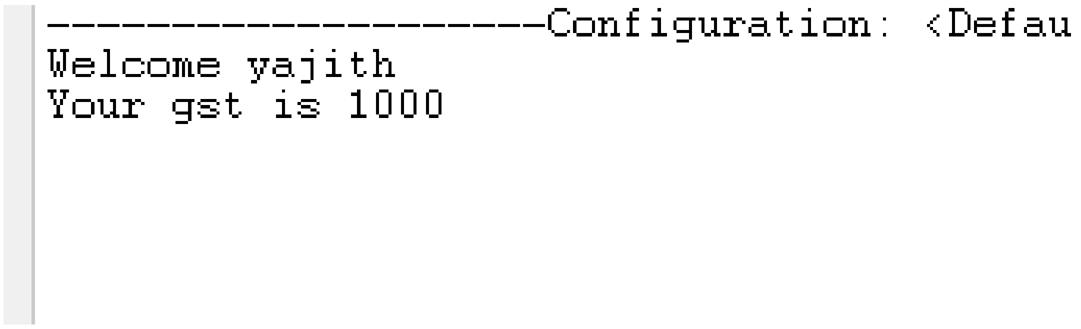
Server

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

public class ServerEcho
{
public static int gst(int amount,int per)
{
int result=(amount*per)/100;
return result;
}
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];
byte[] receive1 = new byte[65535];
int[] per=new int[]{10,5,14};
DatagramPacket DpReceive = null;
DatagramPacket DpReceive1 = null;
int p=1;
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);
String inp = new String(receive, 0, receive.length);
String[] arr=inp.split(" ",6);
System.out.println("Welcome "+arr[0]);
int amount=Integer.parseInt(arr[1].trim());
int sel=Integer.parseInt(arr[2].trim());
if(sel==1)
```

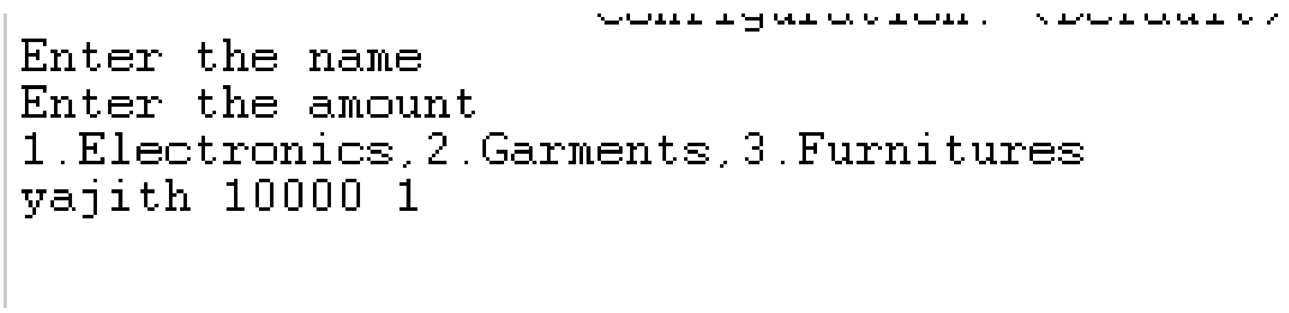
```
{
p=per[0];
}
else if(sel==2)
{
p=per[1];
}
else
{
p=per[2];
}
int gs=gst(amount,p);
System.out.print("Your gst is "+gs);

// 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;
}
}
```

SCREEN SHOTS:



```
--------------------Configuration: <Defau
Welcome yajith
Your gst is 1000
```



```
Enter the name
Enter the amount
1.Electronics,2.Garments,3.Furnitures
yajith 10000 1
```

RESULT:

Thus the programs for find the gst for different products are implemented in Java and the results are verified.

Evaluation:

Parameter	Max Marks	Marks Obtained
Complexity of the Application chosen	3	
Uniqueness of the Code	10	
Use of Comment lines and standard coding practices	2	
Viva	5	
Sub Total	20	
Completion of experiment on time	3	
Documentation	7	
Sub Total	10	
Signature of the faculty with Date		