Expt. No. 1 Page No.

Expt. Name. Print - IP Date : 26/07/2021

Aim: A simple program to print the IP address of the system.

Algorithm:

1. Import all the req. packages net and io.
2. Create a class ipclient
3. Initialize InetAddress class and create the object ia.
4. Get the Ip address of the system using getLocalHost() method
5. Print IP address

Program:

```
import java.net.*;
import java.io.*;
class ipclient
{
    public static void main(String args[])
```

Expt. No. _____ Page No. _____

Expt. Name. _____ Date : _____

```
{
    try
    {
        InetAddress ia= InetAddress.getLocalHost();
        System.out.println("The client system address
                            is:"+ia);
    }
    catch (IOException e)
    {
        System.io.println("The exception is:"+e);
    }
}
}
```

Result: The program is executed successfully.

```
bphar@HBP-PC MINGW64 /d/AAdityAA/SIST/Sem 5/Computer Network Lab (master)
$  cd "d:\\AAdityAA\\SIST\\Sem 5\\Computer Network Lab" ; /usr/bin/env c:\\U
cher.bat "C:\\Program Files\\AdoptOpenJDK\\jdk-8.0.292.10-hotspot\\bin\\java
workspaceStorage\\61d1e61acbab636e83af4e66e9e2fc1e\\redhat.java\\jdt_ws\\Com
The client System IP address is:HBP-PC/127.0.0.1
```

② Creation of Date and Time server

Algorithm

Server Algorithm:-

1) Import the necessary packages

2) Create a class DateServer

3) Initialize the classes ServerSocket, PrintStream, DataInputStream, String with objects ss, s, ps, dis, inet respectively

4) Get the ServerSocket port and date

5) Print the date and close server

Program:

```
import java.io.*;
import java.net.*;
import java.ioutil.*;
class DateServer
{
    public static void main(String args[])
```

Expt. No. _____ Page No. _____

Expt. Name. _____ Date : _____

```
{
    ServerSocket ss;
    Socket s;
    PrintStream ps;
    DataInputStream dis;

    String cnet;

    try {

        ss = new ServerSocket(8020);

        while(true)
        {
            s = ss.accept();

            ps = new PrintStream(s.getOutputStream());

            Date d = new Date();
            ps.println
            ps.println(d);
            ps.close();
        }
    } catch (IOException e) {
    System.out.println("Exception in : " + e);
    }
}
```

Result: The program was executed successfully

Algorithm Client prog:

1) Same as DateServer Initialize all classes mentioned there and create the same objects.

2) Using ia object get the Local Host

3) Get sdate using readLine

4) display sdate

Program:

```
import java.io.*;

Import java.net.*;

public class DateClient {
{
    public static voi void main (String args[])
    {

        Socket soc;

        DataInputStream din;

        String sdate;

        PrintStream ps;
```

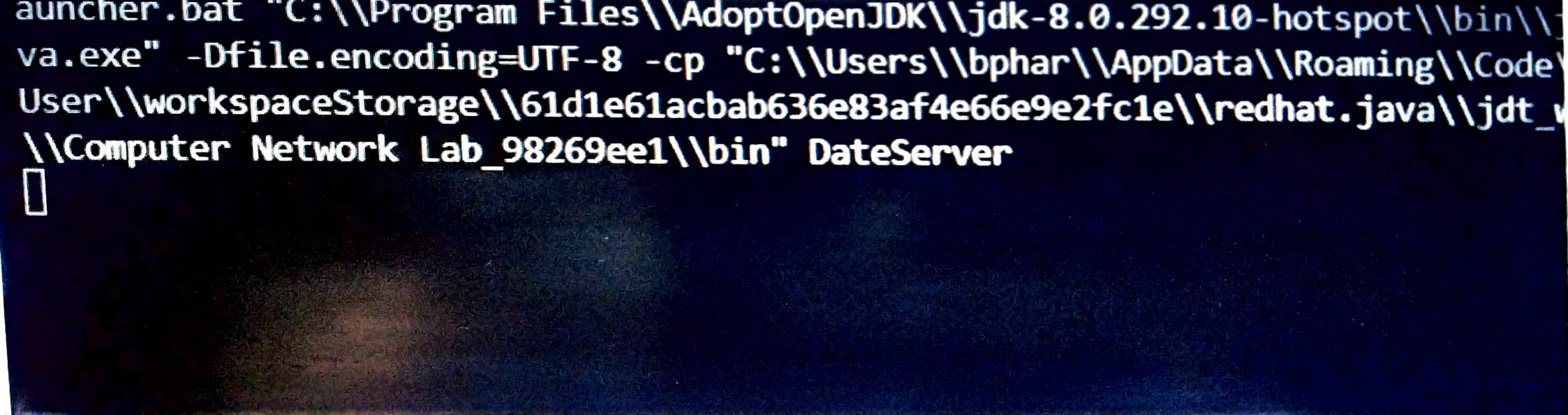
Expt. Name

Date :

```
try {
    InetAddress ia = InetAddress.getLocalHost();
    soc = new Socket(ia, 8020);
    dis = new DataInputStream(soc.getInputStream());

    sdate = dis.readline();

    System.out.println("The date in the server is:"
                        + sdate);

} catch (IOException e) {

    System.out.println("Exception is:"+e);

}
}
}
```

Result: Date of the server displayed successfully.

```
$ java DateClient
The data in the server is: Tue Jul 27 20:32:13 IST 2021
```


auncher.bat "C:\\Program Files\\AdoptOpenJDK\\jdk-8.0.292.10-hotspot\\bin\\j
va.exe" -Dfile.encoding=UTF-8 -cp "C:\\Users\\bphar\\AppData\\Roaming\\Code\
User\\workspaceStorage\\61d1e61acbab636e83af4e66e9e2fc1e\\redhat.java\\jdt_w
\\Computer Network Lab_98269ee1\\bin" DateServer

(3) Printing Client Address at Server Side

Aim: Design a socket program to print the client address at the server side.

Algorithm:

Server:

1) Import all req. packages
2) Create a class ServerTP
3) Initialize the classes ServerSocket and Socket and their respective objects
4) Initialize the class DataInputStream for reading the data from socket through getInputStream() method.
5) Establish the connection with client system using the client IP and port no. 8020
6) Read the IP address from the socket
7) Print the IP address at the server

Client

1) Import all req. packages

Expt. No. _____ Page No. _____

Expt. Name. _____ Date : _____

2) Create class ClientSP

3) Initialize Socket and respective objects

4) Initialize the PrintStream class for writing data through getOutputStream() method

5) Get IP address of system using InetAddress with getLocalHost().

6) Establish the connection with server system IP address and port no. 8020

7) Write IP into Socket

8) If any exception, print error

Program

Server

```
import java.io.*;
import java.net.*;
class SerwerSP
{
    public static void main (String args[]) {
```

```
        ServerSocket ss;
        Socket s;
        DataInputStream dis;
        String ip;

        try {

            ss = new ServerSocket (8020);
            while (true) {
                s = ss.accept();
                dis = new DataInputStream (s.getInputStream());
                ip = dis.readLine();
                System.out.println (" Ip address of the
                        client System : " + ip);
            }
        } catch (IOException e) {
            System.out.println ("Exception in : " + e);
        }
    }
}
```

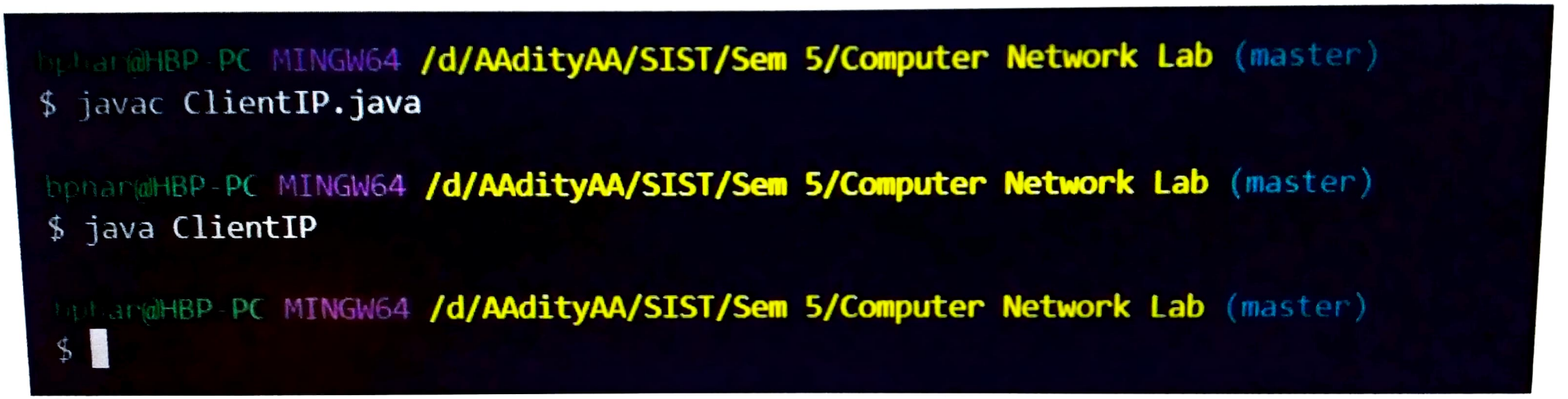
Client

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

class ClientIp {
    public static void main (String args[]) {
        Socket soc;
        PrintStream ps;
        try {
```

Expt. Name: _____ Date : _____

```
            InetAddress ia = InetAddress.getLocalHost();
            soc = new Socket(ia, 8020);
            ps = new PrintStream(soc.getOutputStream());
            ps.println(ia);
        } catch(IOException e) {
            System.out.println("Exception is: "+e);
        }
    }
}
```

Result: Server and client address is printed successfully.


bpnar@HBP-PC MINGW64 /d/AAdityAA/SIST/Sem 5/Computer Network Lab (master)
$ javac ClientIP.java
bpnar@HBP-PC MINGW64 /d/AAdityAA/SIST/Sem 5/Computer Network Lab (master)
$ java ClientIP
bpnar@HBP-PC MINGW64 /d/AAdityAA/SIST/Sem 5/Computer Network Lab (master)
$

```
bphar@HBP-PC MINGW64 /d/AAdityAA/SIST/Sem 5/Computer Network Lab (master)
$ javac ServerIP.java
Note: ServerIP.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.

bphar@HBP-PC MINGW64 /d/AAdityAA/SIST/Sem 5/Computer Network Lab (master)
$ java ServerIP
Ip Address client system: HBP-PC/192.168.0.7
```

④ Creation of UDP Server

Aim: To perform a java prog for UDP client and server

Algorithm

Server:

1) Create a new Datagram Socket and Packet
2) Create message to be sent
3) Convert into bytes & create a packet
4) Send packet
5) wait for acknowledgement from client
6) Print data from client
7) Stop

Client

1) Create new Datagram Socket and Packet
2) get the packet
3) Print Content
4) Create new packet
5) Send to server
6) Stop.

Expt. No. _____ Page No. _____

Expt. Name. _____ Date : _____

Program:

Server

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

public class udpserver {
    public static void main(String args[]) {
        String s;
        InetAddress id = InetAddress.getLocalHost();
        BufferedReader dis = new BufferedReader(
                                new InputStreamReader(System.in));
        DatagramSocket ds = new DatagramSocket(server);

        byte b[] = new byte[1024];
        System.out.println("Server Side ... Sending...");
        System.out.println("\n"+id);
        while(true) {
            s=dis.readLine();
            if(s.equals("end")) {
                b = s.getBytes();
                DatagramPacket dp= new Datagram(
                                    b, s.length(), id, client);
                ds.send(dp);
                break;
            }
            else {
                b = s.getBytes();
```

```
        DatagramPacket dp=new DatagramPacket(
                        b,s.length(),id,client);

                ds.send(dp);
            }
        }
    }
```

client

```
import java.net.*;
import java.io.*;
public class udpclient {

    public static void main(String args[]) {

        DatagramSocket ds= new DatagramSocket(client);

        byte b[] = new byte[1024];

        System.out.println("client ... receiving..");

        while (true)
        {
            DatagramPacket dp = new DatagramPacket
                                (b,b.length);
            ds.receive(dp);
            String s= new String(dp.getData(),0,
                                dp.getLength());
            dp.
```

Expt. No. _____ Page No. _____

Expt. Name. _____ Date : _____

```
if (s.equals("end"))
        break;
else
    System.out.println(s);
}
}
}
```

Result: Udp Server and client executed successfully

```
bphar@HBP-PC MINGW64 /d/AAdityAA/SIST/Sem 5/Computer Network Lab (master)
$ java udpClient
Client... Recieving...
```

```
bphar@HBP-PC MINGW64 /d/AAdityAA/SIST/Sem 5/Computer Network Lab (mas
$ java udpServer
Server Side.. Sending..

HBP-PC/192.168.0.7
```

⑤ chat prog

Aim: ~~For~~ Program a java prog for creation of simple chat

Algorithm

Server

1) Declare variable & structure for socket
2) Socket is binded at the specific port
3) If binding successfull write message to client
4) close socket
5) Execute the program

Client

1) Declare the variables
2) Socket is created & connected
3) Message from server is responded by client

Program

client

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

public class chatClient{
    public static void main(String args[]) {
```

Expt. No. _____ Page No. _____

Expt. Name. _____ Date : _____

throws

```
Socket sk = new Socket(InetAddress.getLocalHost(),2000);

BufferedReader sin = new BufferedReader(new
                        InputStreamReader(System.in));
String s;
while (true) {
    System.out.println("Client: ");
    s = stdin.readLine();
    sout.println(s);
    s = sin.readLine();
    System.out.println("Server: "+s+"\n");
    if ( s.equalsIgnoreCase("BYE") )
        break;
}
sk.close();
sin.close();
sout.close();
stdin.close();
}
}
```

Server

```
import java.io.*;
import java.net.*;
public class chatServer {
    public static void main(String args[]) {
```

```
ServerSocket ss = new ServerSocket(2000);
Socket sk = ss.accept();
BufferedReader cin = new BufferedReader(
                        new InputStreamReader
                        (sk.getInputStream()));
PrintStream cout = new PrintStream(sk.getOutputStream
                                   ());
BufferedReader stdin = new BufferedReader
                       (new InputStreamReader
                       (System.in));
String s;
while (true) {
    s = cin.readLine();
    if (s.equalsIgnoreCase("End")) {
        cout.println("BYE");
        break;
    }
    System.out.println("Client: "+s+"\n");
    System.out.println("Server ");
    s = stdin.readLine();
    cout.println(s);
}
```

Expt. No. _____ Page No. _____

Expt. Name. _____ Date : _____

```
ss.close();
sk.close();
cin.close();
cout.close();
stdin.close();
}
}
```

Result: Chat program implemented Sucessfully.

```
bphar@HBP-PC MINGW64 /d/AAdityAA/SIST/Sem 5/Computer Network Lab (master)
$ javac chatClient.java

bphar@HBP-PC MINGW64 /d/AAdityAA/SIST/Sem 5/Computer Network Lab (master)
$ java chatClient
Client : Hi
Server : Hello
Client : How are You?
Server : I'm fine
Client : good bye
Server : bye
```

```
bphar@HBP-PC MINGW64 /d/AAdityAA/SIST/Sem 5/Computer Network Lab (master)
$ javac chatServer.java

bphar@HBP-PC MINGW64 /d/AAdityAA/SIST/Sem 5/Computer Network Lab (master)
$ java chatServer
Client : Hi
Server : Hello
Client : How are You?
Server : I'm fine
Client : good bye
Server : bye
```