NETWORKING LAB

IPCLIENT

IPCLIENT :

```
import java.net.*;
import java.io.*;
import java.util.*;
public class ipclient {
   public static void main(String[] args) {
      try {
         InetAddress ia=InetAddress.getLocalHost();
         System.out.println("IP Adress is : "+ia);
      }
      catch(IOException except) {
         System.out.println("The exception is :
"+except);
      }
   }
}
```

DATESERVER AND DATECLIENT

DATESERVER :

```
import java.io.*;
import java.net.*;
import java.util.*;
class dateserver
{
public static void main(String args[])
{
ServerSocket ss;
Socket s;
PrintStream ps;
DataInputStream dis;
String inet;
try
{
ss=new ServerSocket(8020);
while(true)
{
s=ss.accept();
ps=new PrintStream(s.getOutputStream());
Date d=new Date();
ps.println(d);
dis=new DataInputStream(s.getInputStream());
inet=dis.readLine();
System.out.println("IP Address of the client is : "+inet);
ps.close();
}
}
catch(IOException e)
{
System.out.println("The exception is: "+e);
}  }  }
```

DATECLIENT :

```
import java.io.*;
import java.net.*;
class dateclient
{
 public static void main(String args[])
 {
 Socket soc;
 DataInputStream dis;
 String sdate;
 PrintStream ps;
  try
  {
  InetAddress ia=InetAddress.getLocalHost();
  soc=new Socket(ia,8020);
  ps=new PrintStream(soc.getOutputStream());
  dis=new DataInputStream(soc.getInputStream());
  sdate=dis.readLine();
  System.out.println("The date in the server is:
"+sdate);
  ps.println(ia);
  ps.close();
  }
  catch(IOException e)
  {
  System.out.println("The exception is: "+e);
  }
 }
}
```

ECHOSERVER AND ECHOCLIENT

ECHOSERVER :

```
import java.io.*;
import java.net.*;
public class EchoServer
{
public EchoServer(int portnum)
{
try
{
server = new ServerSocket(portnum);
}
catch (Exception err)
{
System.out.println(err);
}
}
public void serve()
{
try
{
while (true)
{
Socket client = server.accept();
BufferedReader r = new BufferedReader(new
InputStreamReader(client.getInputStream()));
PrintWriter w = new PrintWriter(client.getOutputStream(),true);
w.println("Welcome to the Java EchoServer. Type 'bye'to close.");
String line;
do
```

```
{
line = r.readLine();
if ( line != null )
w.println("Got: "+ line);
System.out.println (line);
}
while ( !line.trim().equalsIgnoreCase("bye") );
client.close();
}
}
catch (Exception err)
{
System.err.println(err);
}
}
public static void main(String[] args)
{
EchoServer s = new EchoServer(9999);
s.serve();
}
private ServerSocket server;
}
```

ECHOCLIENT :

```
import java.io.*;
import java.net.*;
public class EchoClient
{
public static void main(String[] args)
{
try
{
Socket s = new Socket("127.0.0.1", 9999);
BufferedReader r = new BufferedReader(new
InputStreamReader(s.getInputStream()));
PrintWriter w = new
PrintWriter(s.getOutputStream(), true);
BufferedReader con = new BufferedReader(new
InputStreamReader(System.in));
String line;
do
{
line = r.readLine();
if ( line != null )
System.out.println(line);
line = con.readLine();
w.println(line);
}
while ( !line.trim().equalsIgnoreCase("bye") );
}
catch (Exception err)
{
System.err.println(err);
}
}
}
```

CHATSERVER AND CHATCLIENT

CHATSERVER :

```
import java.net.*;
import java.io.*;
public class chatserver{
	public static void main(String args[]) throws Exception{
		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("Bye")){
				cout.println("BYE");
				break;
			}
			System.out.print("Client:"+s+"\n");
			System.out.print("Server:");
			s=stdin.readLine();
			cout.println(s);
		}
		ss.close();
		sk.close();
		cin.close();
		cout.close();
		stdin.close();
	}
}
```

CHATCLIENT :

```
import java.net.*;
import java.io.*;
public class chatclient{
public static void main(String args[]) throws Exception{
Socket sk=new Socket("127.0.0.1",2000);
BufferedReader sin=new BufferedReader(new
InputStreamReader(sk.getInputStream()));
PrintStream sout=new PrintStream(sk.getOutputStream());
BufferedReader stdin=new BufferedReader(new
InputStreamReader(System.in));
String s;
while(true){
System.out.print("Client:");
s=stdin.readLine();
 sout.println(s);
s=sin.readLine();
System.out.print("Server:"+s+"\n");
if(s.equalsIgnoreCase("BYE")){
sout.println("BYE");
break;
}
}
sk.close();
sin.close();
sout.close();
stdin.close();
}
}
```

FILESERVER AND FILECLIENT

FILESERVER :

```
import java.net.*;
import java.io.*;
public class FileServer {
   public static void main(String[] args) throws IOException {
      ServerSocket serverSocket=null;
      try{
         serverSocket=new ServerSocket(8888);
      }
      catch(IOException e){
         System.err.println("Could not listen on port:8888.");
         System.exit(1);
      }
      Socket clientSocket=null;
      try{
         System.out.println("Waiting for connection...");
         clientSocket=serverSocket.accept();
         System.out.println("Accepted
connection:"+clientSocket);
      }
      catch(IOException e){
         System.err.println("Accept failed.");
         System.exit(1);
      }
```

```
        InputStream
in=clientSocket.getInputStream();
        OutputStream out=new
FileOutputStream("recieved_file.txt");
        byte[] bytes=new byte[1024];
        int count;
        while((count=in.read(bytes))>0){
            out.write(bytes,0,count);
        }
        out.close();
        in.close();
        clientSocket.close();
        serverSocket.close();
    }
}
```

FILECLIENT :

```
import java.net.*;
import java.io.*;
public class FileClient {
   public static void main(String[] args) throws
IOException {
      Socket socket=null;
      try{
         socket=new Socket("localhost",8888);
      }
      catch(UnknownHostException e){
         System.err.println("Unknown host:localhost.");
         System.exit(1);
      }
      catch(IOException e){
         System.err.println("Could not connect to
localhost.");
         System.exit(1);
      }
      File file=new File("file_to_send.txt");
      FileInputStream in=new FileInputStream(file);
      OutputStream out=socket.getOutputStream();
      byte[]bytes=new byte[1024];
      int count;
```

```
        while((count=in.read(bytes))>0){
            out.write(bytes,0,count);
        }
        out.close();
        in.close();
        socket.close();
    }
}
```

UDPSERVER AND UDPCLIENT

UDPSERVER :

```
import java.net.*;
import java.io.*;
public class UDPServer {
   public static void main(String[] args)throws
IOException {
      byte b[] = new byte[2048];
      System.out.println("UDP Server Running....!");
      DatagramSocket dsoc = new
DatagramSocket(1000);
      FileOutputStream fout = new
FileOutputStream("UDPRecieve.txt");
      DatagramPacket dp = new
DatagramPacket(b,b.length);
      dsoc.receive(dp);
      String str = new String(dp.getData());
      fout.write(str.getBytes());
      System.out.println("File transfer
completed....!");
      fout.close();
   }
}
```

UDPCLIENT :

```
import java.net.*;
import java.io.*;
public class UDPClient{
   public static void main(String args[])throws
Exception{
      byte b[] = new byte[1024];
      System.out.println("Connecting UDP
Server....!");
      FileInputStream fin = new
FileInputStream("UDPSend.txt");
      DatagramSocket dsoc = new DatagramSocket();
      int i = 0;
      while(fin.available() != 0){
         b[i] = (byte)fin.read();
         i++;
      }
      fin.close();
      dsoc.send(new
DatagramPacket(b,i,InetAddress.getLocalHost(),10
00));
   }
}
```

PINGIP

PINGIP :

```
import java.io.*;
import java.util.*;
public class pingip {
   public static void runSystemCommand(String
Command){
      try{
         Process p=
Runtime.getRuntime().exec(Command);
         BufferedReader InputStream=new
BufferedReader(new
InputStreamReader(p.getInputStream()));
         String s="";
         while((s=InputStream.readLine())!=null){
            System.out.println(s);
         }
      }
      catch(Exception e){
         e.printStackTrace();
      }
   }
   public static void main(String a[]){
      String ip="localhost";
      runSystemCommand("ping "+ip);
      Date date=new Date();
      System.out.println(date);
   }
}
```

TRACEROUTE

TRACEROUTE :

```
import java.io.*;
import java.util.*;
public class tracert {
   public static void SystemCommand(String
Command){
      try{
         Process
p=Runtime.getRuntime().exec(Command);
         BufferedReader InputStream=new
BufferedReader(new
InputStreamReader(p.getInputStream()));
         String s=" ";
         while((s=InputStream.readLine())!=null){
            System.out.println(s);
         }
      }
      catch(Exception e){
         e.printStackTrace();
      }
   }
   public static void main(String[] args) {
      String Ip="www.google.co.in";
      SystemCommand("tracert "+Ip);
      Date date=new Date();
      System.out.println(date);
   }
}
```

STOPWAIT PROTOCOL

STOPWAITRECEIVER :

```
import java.io.*;
import java.net.*;
class stopwaitreceiver
{
   public static void main(String args[]) throws Exception
   {
      stopwaitreceiver swr = new stopwaitreceiver();
      swr.run();
   }
   public void run() throws Exception
   {
      String temp="any message", str="exit";
      ServerSocket myss=new ServerSocket(9999);
      Socket ss_accept=myss.accept();
      BufferedReader ss_bf=new BufferedReader(new
InputStreamReader(ss_accept.getInputStream()));
      PrintStream myps=new
PrintStream(ss_accept.getOutputStream());
      while(temp.compareTo(str)!=0)
      {
         Thread.sleep(1000);
         temp=ss_bf.readLine();
         if(temp.compareTo(str)==0)
         {
            break;
         }
```

```
            Thread.sleep(500);
            myps.println("Received");
        }
        System.out.println("ALL FRAMES WERE
RECEIVED SUCCESSFULLY");
    }
}
```

STOPWAITSENDER :

```
import java.io.*;
import java.net.*;
import java.util.Scanner;
class stopwaitsender
{
   public static void main(String args[]) throws Exception
   {
      stopwaitsender sws=new stopwaitsender();
      sws.run();
   }
   public void run() throws Exception
   {
      Scanner sc=new Scanner(System.in);
      System.out.println("Enter no of frames to be sent");
      int n=sc.nextInt();
      Socket myskt=new Socket("localhost", 9999);
      PrintStream myps=new
PrintStream(myskt.getOutputStream());
      for(int i=0;i<=n;i++)
      {
         if(i==n)
         {
            myps.println("exit");
            break;
         }
         System.out.println("Frame no "+i+" is sent");
         myps.println(i);
```

```
        BufferedReader bf=new
BufferedReader(new
InputStreamReader(myskt.getInputStream()));
        String ack=bf.readLine();
        if(ack!=null)
        {
           System.out.println("Acknowledgement
was Received from receiver");
           Thread.sleep(4000);
        }
        else
        {
           myps.println(i);
        }
     }
   }
}
```

DNS

DNS :

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

public class DNS
{
   public static void main(String[] args)
   {
      int n;
      BufferedReader in = new BufferedReader(new
InputStreamReader(System.in));
      do
      {
         System.out.println("\n Menu: \n 1. DNS 2. Reverse DNS 3.
Exit \n");
         System.out.println("\n Enter your choice");
         n = Integer.parseInt(System.console().readLine());
         if(n==1)
         {
            try
            {
               System.out.println("\n Enter Host Name ");
               String hname=in.readLine();
               InetAddress address;
               address = InetAddress.getByName(hname);
```

```
                System.out.println("Host Name:" +
address.getHostName());
                System.out.println("IP:" +
address.getHostAddress());
            }
            catch (IOException ioe)
            {
                ioe.printStackTrace();
            }
        }
        if(n==2)
        {
            try
            {
                System.out.println("\n Enter IP
address");
                String ipstr = in.readLine();
                InetAddress ia =
InetAddress.getByName(ipstr);
                System.out.println("IP: "+ipstr);
                System.out.println("Host Name:"
+ia.getHostName());
            }
            catch (IOException ioe)
            {
                ioe.printStackTrace();
            }
        }
    }
    while (!(n==3));
  }
}
```

HTTP PROTOCOL

HTTP :

```
import java.io.*;
import java.net.*;
public class http
{
public static void main(String args[])throws
IOException
{
URL url=new URL("https://www.google.co.in/");
URLConnection conn=url.openConnection();
conn.connect();
InputStreamReader content= new
InputStreamReader(conn.getInputStream());
FileWriter f=new FileWriter ("abc.html");
for(int i=0;i!=-1;i= content.read())
{
f.write((char) i);
}
}
}
```

CALCULATION OF CHECKSUM

CHECKSERVER :

```
import java.io.*;
import java.net.*;
import java.util.zip.*;
public class CheckServer {
   public static void main(String[] args) throws Exception {
      ServerSocket serverSocket = new ServerSocket(1234);
      System.out.println("Server started");
      while (true) {
         Socket clientSocket = serverSocket.accept();
         System.out.println("Client connected: " +
clientSocket.getInetAddress().getHostAddress());
         InputStream inputStream =
clientSocket.getInputStream();
         BufferedInputStream bufferedInputStream = new
BufferedInputStream(inputStream);
         byte[] data = bufferedInputStream.readAllBytes();
         Checksum checksum = new CRC32();
         checksum.update(data, 0, data.length);
         long checksumValue = checksum.getValue();
         System.out.println("Checksum value of received data: " +
checksumValue);
         FileOutputStream fileOutputStream = new
FileOutputStream("receivedData.txt");
         fileOutputStream.write(data);
         fileOutputStream.close();
         clientSocket.close();
      }
   }
}
```

CHECKCLIENT :

```
import java.io.*;
import java.net.*;
import java.util.zip.*;
public class CheckClient {
    public static void main(String[] args) throws Exception {
        Socket socket = new Socket("localhost", 1234);
        System.out.println("Connected to server");
        OutputStream outputStream = socket.getOutputStream();
        BufferedOutputStream bufferedOutputStream = new
BufferedOutputStream(outputStream);
        FileInputStream fileInputStream = new
FileInputStream("dataToSend.txt");
        byte[] data = fileInputStream.readAllBytes();
        fileInputStream.close();
        bufferedOutputStream.write(data);
        bufferedOutputStream.flush();
        Checksum checksum = new CRC32();
        checksum.update(data, 0, data.length);
        long checksumValue = checksum.getValue();
        System.out.println("Checksum value of sent data: " +
checksumValue);
        socket.close();
    }
}
```