COMPUTER NETWORKS(Theory and Practical)

MANUAL

//1. A PROGRAM TO FIND AN ADDRESS OF IP

```
import java.io.*;
import java.net.*;
class address1
{
public static void main(String arg[])
{
try
{
InetAddress address = InetAddress.getLocalHost();
System.out.println(address);
}
catch(UnknownHostException e)
{
System.out.println("could not find this computer address");
}
}
}
```

OUTPUT

```
D:\jdk1.6\bin>javac address1.java

D:\jdk1.6\bin>java address1

M35/192.168.21.115
```

//2. A PROGRAM TO FIND THE HOST NAME OF THE LOCAL MACHINE

```
import java.io.*;
import java.net.*;
class myname
{
public static void main(String arg[])
{
try
{
InetAddress address = InetAddress.getLocalHost();
System.out.println("Hello my host name is " + address.getHostName( ));
}
catch(UnknownHostException e)
{
System.out.println("I'm sorry, i dont know my own name");
}
}
}
```

OUTPUT

```
D:\jdk1.6\bin>javac myname.java

D:\jdk1.6\bin>java myname

Hello my host name is M35
```

//3. RETRIEVING DATA WITH URL

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

class Url
{
	public static void main(String arg[])throws Exception
	   {
	      int c;
	      URL hp=new URL("file://d:/exam/Hai.txt");
	     URLConnection hpcon=hp.openConnection();
	     System.out.println("Date:"+ new Date(hpcon.getDate()));
	     System.out.println("Content-Type:"+ hpcon.getContentType());
	     System.out.println("Expiration:"+ new Date(hpcon.getExpiration()));
	     System.out.println("Last-Modified:"+ new Date(hpcon.getLastModified()));
	     int len=hpcon.getContentLength();
	     System.out.println("Content Length:"+len);
	     if(len>0)
	     {
	           System.out.print("Content");
	           InputStream input=hpcon.getInputStream();
	           int i=len;
	           while(((c=input.read())!=-1) && (--i>0))
	                 System.out.print((char)c);
	           input.close();
	     }
	     else
	           System.out.println("No Content Available.....");
	 }
}
```

INPUT / OUTPUT:

```
Date:Thu Jan 01 05:30:00 IST 1970
Content-Type:text/plain
Expiration:Thu Jan 01 05:30:00 IST 1970
Last-Modified:Mon Feb 13 10:48:52 IST 2010

Content Length:48

Content

Network lab...
Retrieving data with URLs...
```

```
//4. ECHO-CLIENT
import java.io.*;
import java.net.*;
import java.lang.String;
public class EchoClient
{
	public static void main(String arg[]) throws Exception
	{
	DataInputStream in=new DataInputStream(System.in);
	Socket s=new Socket("localhost",1234);
	DataInputStream inecho=new DataInputStream(s.getInputStream());
	DataOutputStream out=new DataOutputStream(s.getOutputStream());
	String str;
	System.out.println("\nclient side:");
	System.out.println("\ntype exit to quit");
	System.out.println("\nEnter the client message:");
	while(true)
	{
	str=in.readLine();
	if(str.equals("exit"))
	{
	out.writeBytes("exit");
	break;
	}
	else
	{
	out.writeBytes(str+"\n");
	System.out.println("Echo from server:");
	System.out.println(inecho.readLine());
	System.out.println("\nEnter client message:");
	}
	}
	inecho.close();
	out.close();
	}
	}
```

```
//ECHO-SERVER
import java.io.*;
import java.net.*;
import java.lang.String;
public class EchoServer
{
public static void main(String args[]) throws Exception
{
ServerSocket ss=new ServerSocket(1234);
Socket s=ss.accept();
DataInputStream in=new DataInputStream(s.getInputStream());
DataOutputStream out=new DataOutputStream(s.getOutputStream());
String str;
System.out.println("\nserver side:");
while(true)
{
str=in.readLine();
System.out.println("Msg from client:");
System.out.println(str+"\n");
if(str.equals("exit"))
break;
else
out.writeBytes(str+"\n");
}
in.close();
out.close();
s.close();
}
}
```

INPUT / OUTPUT:

D:\jdk1.6\bin>javac EchoClient.java

D:\jdk1.6\bin>java EchoClient

Echo -client side:

Type exit to quit

Enter the client message:
Hi, Goodmorning

Echo from server:

Hi, Goodmorning

Enter client message:
This is text msg from client

Echo from server:
This is text msg from client

Enter client message:
Exit

Echo from server:
Exit

Echo -Server side:

D:\jdk1.6\bin>javac EchoServer.java

D:\jdk1.6\bin>java EchoServer

server side:
Msg from client:
Hi, Goodmorning

Msg from client:
This is text msg from client

Msg from client:
Exit

//5. PING – CLIENT:

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

public class pingClient
{
	public static void main(String args[]) throws Exception
	{
		String str;
		int c=0;
		long t1,t2;

			Socket s = new Socket("localhost",555);
			DataInputStream in=new DataInputStream(s.getInputStream());
			PrintStream out = new PrintStream(s.getOutputStream());
			while(c<4)
			{
				t1 = System.currentTimeMillis();
				str="Welcome to vinayaka mission College";
				out.println(str);
				System.out.print(in.readLine());
				t2 = System.currentTimeMillis();
				System.out.println("; TTL = " + (t2 - t1) + " ms");
				c++;
			}
			s.close();

	}

}
```

// PING - SERVER:

```
import java.io.*;
import java.net.*;
import java.util.*;
import java.text.*;
class pingServer
{
	public static void main(String args[]) throws Exception
	{
		ServerSocket ss=new ServerSocket(555);
		Socket s = ss.accept();
		int c=0;
		while(c<4)
		{
		DataInputStream in = new DataInputStream(s.getInputStream());
		PrintStream out = new PrintStream(s.getOutputStream());
		String str = in.readLine();
		out.println("Reply from " + InetAddress.getLocalHost() + "; Length " +
		str.length());
		c++;
		}
		s.close();
	}
}
```

INPUT / OUTPUT:

PING – CLIENT:

D:\jdk1.6\bin>javac pingClient.java

D:\jdk1.6\bin>java pingClient
Reply from M35/192.168.21.115; Length 35; TTL = 31 ms
Reply from M35/192.168.21.115; Length 35; TTL = 0 ms
Reply from M35/192.168.21.115; Length 35; TTL = 0 ms
Reply from M35/192.168.21.115; Length 35; TTL = 0 ms

PING – SERVER:

D:\jdk1.6\bin>javac pingServer.java

D:\jdk1.6\bin>java pingServer

//6. TALK-CLIENT

```
import java.io.*;
import java.net.*;
import java.lang.String;
public class talkc
{
public static void main(String args[]) throws IOException
{
DataInputStream in=new DataInputStream(System.in);
Socket s=new Socket("localhost",1235);
DataInputStream inp=new DataInputStream(s.getInputStream());
DataOutputStream out=new DataOutputStream(s.getOutputStream());
String str;
System.out.println("\n Client side :");
System.out.println("\n Type exit to quit");
while(true)
{
System.out.println("\n Enter the client message :");
str=in.readLine();
if(str.equals("exit"))
{
out.writeBytes("exit");
break;
}
else
{
out.writeBytes(str+"\n");
System.out.println("Reply from server :");
System.out.println(inp.readLine());
}
}
inp.close();
out.close();
}
}
```

//TALK-SERVER

```
import java.io.*;
import java.net.*;
import java.lang.String;
public class talks
{
public static void main(String args[]) throws Exception
{
DataInputStream in=new DataInputStream(System.in);
ServerSocket ss=new ServerSocket(1235);
Socket s=ss.accept();
DataInputStream inp=new DataInputStream(s.getInputStream());
DataOutputStream out=new DataOutputStream(s.getOutputStream());
String str,str1;
System.out.println("\n server side :");
while(true)
{
str=inp.readLine();
System.out.println("From Client : ");
System.out.println(str+"\n");
if(str.equals("exit"))
break;
else
{
System.out.println("Enter server message :");
str1=in.readLine();
out.writeBytes(str1+"\n");
}
}
in.close();
out.close();
s.close();
}
}
```

INPUT / OUTPUT:

TALK – CLIENT:

```
D:\jdk1.6\bin>javac talkc.java

D:\jdk1.6\bin>java talkc
Client side:
Type exit to quit

Enter the client message:
Hi, How r u?
Reply from server :
I'm fine

Enter the client message :
Convey my regards to ur family
Reply from server :
Sure

 Enter the client message :
 Exit
```

TALK – SERVER:

```
D:\jdk1.6\bin>javac talks.java

D:\jdk1.6\bin>java talks

Server side:
From Client:
Hi, How r u?

Enter server message:
I'm fine
From Client:
Convey my regards to ur family

Enter server message:
Sure

From Client:
Exit
```

// 7. TCP – CLIENT APPLET

```
import java.io.*;
import java.net.*;
import java.awt.*;
import java.applet.Applet;
import java.awt.event.*;
/*<applet code = "clientapplet" width = 300 height = 200>
</applet>*/
public class clientapplet extends Applet implements ActionListener
{
Button area;
TextField rad;
Label l;
double result;
String result1 = " ";
InputStream is;
DataInputStream dis;
OutputStream os;
DataOutputStream dos;
public void init( )
{
area = new Button ("Area");
rad = new TextField (5);
l = new Label ("Enter the Radius");
add(l);
add(rad);
add(area);
area.addActionListener(this);
}
public void actionPerformed (ActionEvent ae)
{
try
{
Socket s = new Socket ("127.0.0.1", 9999);
is = s.getInputStream( );
dis = new DataInputStream(is);
os = s. getOutputStream( );
dos = new DataOutputStream(os);
if (ae.getSource ( ) == area)
{
String str = rad.getText( );
double radius = Double.parseDouble (str);
dos.writeDouble (radius);
result = dis.readDouble( );
result1 = String.valueOf (result);
repaint ( );
```

```
}
}
catch (IOException ioe)
{
System.out.println(ioe);
}
}
public void paint (Graphics g)
{
g.drawString("The Area of the given radius is : " + String.valueOf (" "+result1), 10,100);
}
}
```

//TCP – AREASERVER

```
import java.io.*;
import java.net.*;
class areaserver
{
public static void main(String args[])
{
try
{
int port = Integer.parseInt(args[0]);
ServerSocket ss = new ServerSocket (port);
Socket s = ss.accept();
InputStream is = s.getInputStream();
DataInputStream dis = new DataInputStream (is);
double radius = dis.readDouble( );
OutputStream os = s.getOutputStream( );
DataOutputStream dos = new DataOutputStream (os);
dos.writeDouble (3.14 * radius * radius);
s.close();
}
catch (Exception e)
{
System.out.println("Exception:" + e);
}
}
}
```

INPUT / OUTPUT:

TCP – AREASERVER:

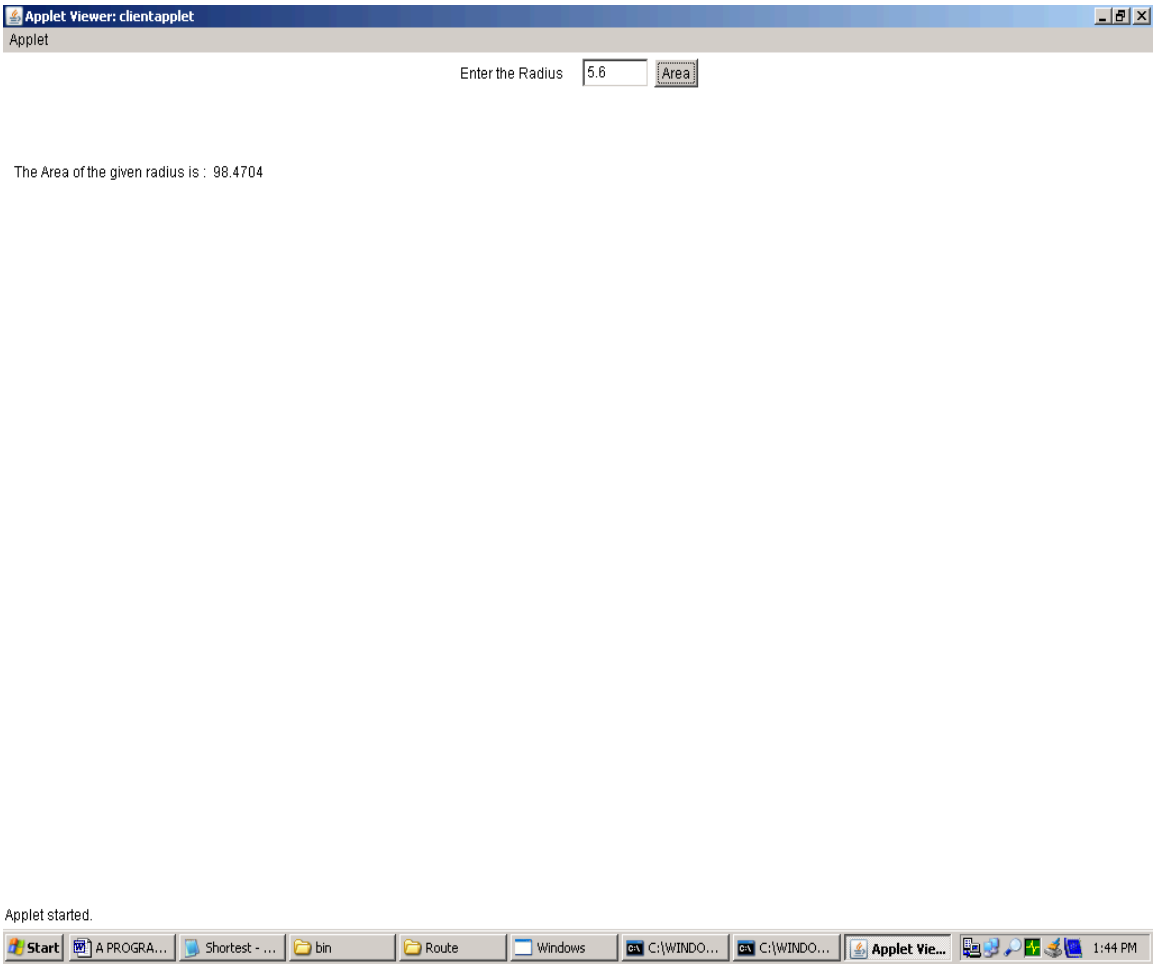
D:\jdk1.6\bin>javac areaserver.java

D:\jdk1.6\bin>java areaserver 6666

TCP – CLIENT APPLET:

D:\jdk1.6.0\bin>javac clientapplet.java

D:\jdk1.6.0\bin>appletviewer clientapplet.java



// 8.UDP – Program to receive a string from the client

```
import java.io.*;
import java.net.*;
class datagramprogram1
{
public static void main(String args[])
{
try
{
int port = Integer.parseInt ("4444");
DatagramSocket ds = new DatagramSocket(port);
byte buffer [ ] = new byte[20];
while (true)
{
DatagramPacket dp = new DatagramPacket (buffer, buffer.length);
ds.receive (dp);
String str = new String(dp.getData( ));
System.out.println(str);
}
}
catch (Exception e)
{
e.printStackTrace( );
}
}
}
```

// UDP – Program to send a string from the server

```
import java.io.*;
import java.net.*;
class datagramprogram2
{
public static void main(String args[])
{
try
{
InetAddress ia = InetAddress.getByName("127.0.0.1");
int port = Integer.parseInt ("4444");
DatagramSocket ds = new DatagramSocket();
String s1 ="hello";
byte buffer [ ] = s1.getBytes( );
DatagramPacket dp = new DatagramPacket (buffer, buffer.length,ia,port);
ds.send (dp);
}
catch (Exception e)
{
System.out.println("Exception:" + e);
}
}
}
```

INPUT / OUTPUT

UDP – Program to receive a string from the client

D:\jdk1.6\bin>javac datagramprogram1.java

D:\jdk1.6\bin>java datagramprogram1

hello

UDP – Program to send a string from the client

D:\jdk1.6\bin>javac datagramprogram2.java

D:\jdk1.6\bin>java datagramprogram2

hello

// 9. FTP - CLIENT

```
import java.io.*;
import java.net.*;
public class FileClient
{
public static void main(String args[]) throws IOException
        {
        Socket s=null;
        DataInputStream si=null;

        s=new Socket("localhost",55555);
        si=new DataInputStream(s.getInputStream());
        DataInputStream inp=new DataInputStream(System.in);
        DataOutputStream so=new DataOutputStream(s.getOutputStream());
        String str;
        System.out.println("\nEnter the filename(path):");
        str=inp.readLine();
        so.writeBytes(str);
        so.writeBytes("\n");

        FileOutputStream  fos =new FileOutputStream("abc.txt");

        int str1;
        while((str1=si.read())!=-1)
                fos.write((char)str1);
        System.out.println("\nFile received successfully!");
        si.close();
        }
}
```

```
//FTP –SERVER
import java.io.*;
import java.net.*;
class FileServer
{
        public static void main(String args[]) throws IOException{
        Socket s=null;
        ServerSocket ss=null;
        DataOutputStream sso=null;
        DataInputStream sin=null;

        ss=new ServerSocket(55555);
        // client socket in server machine
        s=ss.accept();
        sso=new DataOutputStream(s.getOutputStream());
        sin=new DataInputStream(s.getInputStream());

        String s1;
        s1=sin.readLine();

        FileInputStream fos=new FileInputStream(s1);
        int str;
        while((str=fos.read())!=-1)
                sso.writeBytes(""+(char)str);
        System.out.println("File has been sent successfully!");
        sso.close();
        s.close();
        }
    }
```

INPUT / OUTPUT:

FTP - FILE CLIENT

D:\jdk1.6\bin>javac FileClient.java

D:\jdk1.6\bin>java FileClient

Enter the filename (path):

d:\jdk1.6\bin\abc.txt

File received successfully!

FTP – FILE SERVER

D:\jdk1.6\bin>javac FileServer.java

D:\jdk1.6\bin>java FileServer

File has been sent successfully!

//10. ARP – CLIENT:

```
import java.io.*;
import java.net.*;
class arpClient
	{
	public static void main(String a[])throws Exception
		{
		Socket s=new Socket("localhost",555);
		DataInputStream in=new DataInputStream(s.getInputStream());
		DataOutputStream out=new DataOutputStream(s.getOutputStream());
		String
		iparr[]={"172.16.5.20","172.16.5.21","172.16.5.22","172.16.5.23"};
		String
		macarr[]={"01-10-B5-F2-EF-20","02-11-B6-F3-EG-21","03-12-B7-F4-E
		H-22","04-13-B8-F5-EI-23"};
		String str=in.readLine();
		System.out.println("IP Address received from server:"+str);
		int flag=0;
		for(int i=0;i<5;i++)
			{
			if(str.equals(iparr[i])==true)
				{
				flag=1;
					String str1=macarr[i];
					out.writeBytes(str1+"\n");
					break;
				}
		}
		if(flag==0)
			System.out.println("Given IP Address is not in this network");
		s.close();
		}
	}
```

```
//ARP – SERVER
import java.io.*;
import java.net.*;
class arpServer
	{
	public static void main(String arg[]) throws Exception
		{
		ServerSocket ss=new ServerSocket(555);
		Socket s=ss.accept();
		DataInputStream in=new DataInputStream(s.getInputStream());
		DataOutputStream out=new DataOutputStream(s.getOutputStream());
		DataInputStream sysin =new DataInputStream(System.in);
		System.out.print("Enter an ip address :");
		String str=sysin.readLine();
		out.writeBytes(str+"\n");
		System.out.println("The corresponding MAC Address is :"+in.readLine());
		}
	}
```

INPUT / OUTPUT:
ARP – SERVER:

D:\jdk1.6\bin>javac arpServer.java

D:\jdk1.6\bin>java arpServer

Enter an ip address :172.16.5.21

The corresponding MAC Address is : 01-10-B5-F2-EF-20

ARP – CLIENT:

D:\jdk1.6\bin>javac arpClient.java

D:\jdk1.6\bin>java arpClient

IP Address received from server: 172.16.5.21

```
//11. RARP CLIENT
import java.io.*;
import java.net.*;
class RarpClient
	{
	public static void main(String a[])throws IOException
		{
		Socket s=new Socket("localhost",555);
		DataInputStream in=new DataInputStream(s.getInputStream());
		DataOutputStream out=new DataOutputStream(s.getOutputStream());
		String
		iparr[]={"172.16.5.20","172.16.5.21","172.16.5.22","172.16.5.23"};
		String
		macarr[]={"01-10-B5-F2-EF-20","02-11-B6-F3-EG-21","03-12-B7-F4-E
		H-22","04-13-B8-F5-EI-23"};
		String str=in.readLine();
		System.out.println("MAC Address received from server:"+str);
		int flag=0;
		for(int i=0;i<5;i++)
			{
			if(str.equals(macarr[i])==true)
				{
					flag=1;
					String str1=iparr[i];
					out.writeBytes(str1+"\n");
					break;
				}
		}
		if(flag==0)
			System.out.println("Given MAC Address is not in this network");
		s.close();
		}
	}
```

//RARP – SERVER

```
import java.io.*;
import java.net.*;
class RarpServer
        {
        public static void main(String arg[]) throws IOException
                {
                ServerSocket ss=new ServerSocket(5555);
                Socket s=ss.accept();
                DataInputStream in=new DataInputStream(s.getInputStream());
                DataOutputStream out=new DataOutputStream(s.getOutputStream());
                DataInputStream sysin =new DataInputStream(System.in);
                System.out.print("Enter a MAC address :");
                String str=sysin.readLine();
                out.writeBytes(str+"\n");
                System.out.println("The corresponding IP Address is :"+in.readLine());
                }
        }
```

INPUT / OUTPUT

RARP – SERVER:

D:\jdk1.6\bin>javac RarpServer.java

D:\jdk1.6\bin>java RarpServer

Enter an MAC address: 01-10-B5-F2-EF-20

The corresponding IP Address is: 172.16.5.21

RARP – CLIENT :

D:\jdk1.6\bin>javac RarpClient.java

D:\jdk1.6\bin>java RarpClient

MAC Address received from server: 01-10-B5-F2-EF-20

//12. RCE – CLIENT

```
import java.io.*;
import java.net.*;
class RCEClient
{
public static void main(String arg[]) throws Exception
{
Socket s=new Socket("localhost",333);
DataInputStream in=new DataInputStream(System.in);
DataOutputStream out=new DataOutputStream(s.getOutputStream());
String str;
System.out.println("\nClient side:");
System.out.println("\nType exit to quit");
System.out.println("Enter  the command to execute in server:");
while(true)
{
str=in.readLine();
if(str.equals("exit"))
{
out.writeBytes("exit"+"\n");
break;
}
else
out.writeBytes(str+"\n");
}
in.close();
out.close();
s.close();
}
}
```

```
//RCE - SERVER
import java.io.*;
import java.net.*;
class RCEServer
{
	public static void main(String args[])throws Exception
	{
		ServerSocket ss=new ServerSocket(333);
		Socket s=ss.accept();
		DataInputStream in=new DataInputStream(s.getInputStream());
		String str;
		System.out.println("Server side: ");
		while(true)
		{
			str=in.readLine();
			if(str.equals("exit"))
			break;
			else
			{
				try
				{
					Runtime r=Runtime.getRuntime();
					Process p=null;
					p=r.exec(str);
				}
				catch(Exception f)
				{}
			}
			System.out.println("Executed command is:"+str);
		}
		in.close();
		s.close();
	}
}
```

INPUT / OUTPUT
RCE – CLIENT:

D:\>cd jdk1.6.0

D:\jdk1.6.0>cd bin

D:\jdk1.6.0\bin>javac RCEClient.java

D:\jdk1.6.0\bin>java RCEClient

Client side:

Type exit to quit

Enter the command to execute in server:

Notepad

RCE – SERVER

D:\>cd jdk1.6.0

D:\jdk1.6.0>cd bin

D:\jdk1.6.0\bin>javac RCEServer.java

D:\jdk1.6.0\bin>java RCEServer
Server side:
Executed command is: Notepad