



**ADVANCED
NETWORKING LAB**

NEIL MATHEW – 6CS4 – Y3305 - A2324710002

Q 1: WAP to read a character and a string from console using buffered reader class.

[SOURCE CODE]

```
import java.io.*;

public class p1
{
    public static void main(String args[]) throws IOException
    {
        BufferedReader ob= new BufferedReader(new
InputStreamReader(System.in));
        char c;
        String s;

        System.out.print("\n Enter the string: ");
        s = ob.readLine();

        System.out.print("\n Enter the character: ");
        c=(char)ob.read();

        System.out.println("\n\n The character is: "+c);
        System.out.print(" The string is: "+s);
    }
}
```

Q 1: WAP to read a character and a string from console using buffered reader class.

[SCREENSHOTS]

```
D:\Java\jdk1.6.0_30\bin\0NETWORKING>
javac p1.java
```

```
D:\Java\jdk1.6.0_30\bin\0NETWORKING>
java p1
```

```
Enter the string: Neil
Enter the character: N
```

```
The character is: N
The string is: Neil
```

Q 2: WAP to find out which of the first 1024 ports seems to be hosting TCP server on a specified host.

[SOURCE CODE]

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

public class p2
{
    public static void main(String args[]) throws UnknownHostException
    {
        String host;

        if(args.length != 1)
            host="localhost";
        else
            host=args[0];

        try
        {
            for(int i=1;i<=1024;i++)
            {
                Socket s=new Socket(host,i);
                System.out.println("The Server is running on " + host + " on port
                number " + i);
            }
        }
        catch(IOException e)
        {
            System.out.println(" I/O Error. Message: "+e.getMessage());
        }
    }
}
```

Q 2: WAP to find out which of the first 1024 ports seems to be hosting TCP server on a specified host.

[SCREENSHOTS]

```
D:\Java\jdk1.6.0_30\bin\ONETWORKING>  
java p2 localhost
```

```
I/O Error. Message: Connection refused: connect
```

Q 3: WAP to print the address and the name of the local machine and of two well known sites.

[SOURCE CODE]

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

public class p3
{
public static void main( String args[]) throws Exception
{

    InetAddress local= InetAddress.getLocalHost();
    System.out.println("\nAddress of Local Host: " + local);

    InetAddress ad=InetAddress.getByName("www.yahoo.com");
    System.out.println("\nIP Address of www.yahoo.com : " + ad);

    InetAddress ads[]=InetAddress.getAllByName("www.google.com");

    for (int i=0; i<ads.length;i++)
    {
        int j=i+1;
        System.out.println("Address " + j + " of www.google.com : "
+ ads[i]);
    }
}
}
```

Q 3: WAP to print the address and the name of the local machine and of two well known sites.

[SCREENSHOTS]

```
D:\Java\jdk1.6.0_30\bin\ONETWORKING>javac p3.java
```

```
D:\Java\jdk1.6.0_30\bin\ONETWORKING>java p3
```

```
Address of Local Host: mw-laptop/10.0.42.165
```

```
IP Address of www.yahoo.com : www.yahoo.com/106.10.170.118
```

```
Address 1 of www.google.com : www.google.com/74.125.236.17
```

```
Address 2 of www.google.com : www.google.com/74.125.236.20
```

```
Address 3 of www.google.com : www.google.com/74.125.236.16
```

```
Address 4 of www.google.com : www.google.com/74.125.236.19
```

```
Address 5 of www.google.com : www.google.com/74.125.236.18
```

Q 4: WAP to create a client server application in which client sends some string to server and the server responds by converting it to uppercase (Remote server n client)

[SOURCE CODE]

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

class p4_server
{
public static void main(String args[])throws Exception
{

ServerSocket ss=new ServerSocket(1001);

System.out.println("Listening...");

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

BufferedReader bin=new BufferedReader(new
InputStreamReader(s.getInputStream()));
String s1= bin.readLine();

if(s1.equalsIgnoreCase("Shutdown"))
{
System.out.println(" Shutting Down. Program Stopped! ");
break;
}

System.out.println("\t#Contact with Client Established.");
String s2= s1.toUpperCase();
DataOutputStream dl=new DataOutputStream(s.getOutputStream());
dl.writeBytes(s2+'\n');
System.out.println("\t#Processed input and output returned.");
System.out.println("\n\nListening...");

}
}
}
```


Q 4: WAP to create a client server application in which client sends some string to server and the server responds by converting it to uppercase (Remote server n client)

[SCREENSHOTS]

Client Side:

```
D:\Java\jdk1.6.0_30\bin\0NETWORKING>
javac p4_client.java

D:\Java\jdk1.6.0_30\bin\0NETWORKING>
java p4_client

Enter a string in lower case: make me bigger

Contacting Server...

The received string is: MAKE ME BIGGER
```

Server Side:

```
D:\Java\jdk1.6.0_30\bin\0NETWORKING>
javac p4_server.java

D:\Java\jdk1.6.0_30\bin\0NETWORKING>
java p4_server

Listening...
    #Contact with Client Established.
    #Processed input and output returned.

Listening...
Shutting Down. Program Stopped!
```

Q 5: WAP to design a client server application to implement one way chatting.

[SOURCE CODE]

SERVER SIDE PROGRAM

```
import java.io.*;
import java.net.*;
public class ChatServer
{
public static void main(String args[])throws
IOException,UnknownHostException
{
String S1,S2;
ServerSocket SS=new ServerSocket(1001);
System.out.println("Receiving Message from the client...");
while(true)
{
Socket S=SS.accept();
BufferedReader bin=new BufferedReader(new
InputStreamReader(S.getInputStream()));
DataOutputStream dl=new DataOutputStream(S.getOutputStream());
S1=bin.readLine();
System.out.println("The Recieved message from the client is");
System.out.println(S1);
}
}
}
```

CLIENT SIDE PROGRAM

```
import java.io.*;
import java.net.*;
public class ChatClient
{
public static void main(String args[])throws
UnknownHostException,IOException
{
String sin,sout;
System.out.println("Enter a message to be send to the server: ");
BufferedReader bclient=new BufferedReader(new
InputStreamReader(System.in));
sin=bclient.readLine();
Socket s=new Socket("10.10.1.85",1001);
DataOutputStream dos=new DataOutputStream(s.getOutputStream());
dos.writeBytes(sin+'\n');
s.close();
}
}
```

Q 5: WAP to design a client server application to implement one way chatting.

[SCREENSHOTS]

Client Side:

```
D:\Java\jdk1.6.0_30\bin\0NETWORKING>
javac p5_client.java

D:\Java\jdk1.6.0_30\bin\0NETWORKING>
java p5_client

Enter a message to be send to the server:
Hello Mr. How you doing?
```

Server Side:

```
D:\Java\jdk1.6.0_30\bin\0NETWORKING>
javac p5_server.java

D:\Java\jdk1.6.0_30\bin\0NETWORKING>
java p5_server

Receiving Message from the client...

The Recieved message from the client is
Hello Mr. How you doing?
```

Q 6: Write a program to create a client server application for sharing system date and time.

[SOURCE CODE]

SERVER SIDE PROGRAM

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

class p6_server
{
public static void main(String args[]) throws
Exception,UnknownHostException
{
ServerSocket s=new ServerSocket(5217);
while(true)
{
System.out.println("Waiting For Connection ...");
Socket soc=s.accept();
DataOutputStream out=new DataOutputStream(soc.getOutputStream());
out.writeBytes("Client This is my Date and time:");
out.writeBytes((new Date()).toString() + "\n");
BufferedReader in=new BufferedReader(new
InputStreamReader(soc.getInputStream()));
System.out.println(in.readLine());
soc.close();
}
}
}
```

CLIENT SIDE PROGRAM

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

class p6_client
{
public static void main(String args[]) throws
Exception,UnknownHostException
{
Socket soc=new Socket(InetAddress.getLocalHost(),5217);
BufferedReader in=new BufferedReader(new
InputStreamReader(soc.getInputStream()));
System.out.println(in.readLine());
DataOutputStream out=new DataOutputStream(soc.getOutputStream());
out.writeBytes("Server This is my Date And Time:" );
out.writeBytes((new Date()).toString() + "\n");
}
}
```

Q 6: Write a program to create a client server application for sharing system date and time.

[SCREENSHOTS]

Client Side:

```
D:\Java\jdk1.6.0_30\bin\0NETWORKING>javac
p6_client.java

D:\Java\jdk1.6.0_30\bin\0NETWORKING>java
p6_client

Client This is my Date and time:Mon Mar 25
22:15:37 IST 2013
```

Server Side:

```
D:\Java\jdk1.6.0_30\bin\0NETWORKING>javac
p6_server.java

D:\Java\jdk1.6.0_30\bin\0NETWORKING>java
p6_server

Waiting For Connection ...

Server This is my Date And Time:Mon Mar 25
22:15:37 IST 2013

Waiting For Connection ...
```

Q 7: Write a program to create a chat server that connects two clients that communicate with server using different ports.

[SOURCE CODE]

SERVER SIDE PROGRAM

```
import java.io.*;
import java.net.*;
public class multiserver1 implements Runnable
{
    ServerSocket ss1,ss2;
    DataOutputStream S2c1,S2c2;
    BufferedReader sfcl,sfc2;
    Thread thread;
    Socket s1,s2;
    String s11,s12;
    public multiserver1()
    {
        try
        {
            ss1=new ServerSocket(1001);
            ss2=new ServerSocket(1002);
        }
        catch(Exception e)
        {
            System.out.print(e);
        }
        thread=new Thread(this);
        thread.start();
    }
    public void run()
    {
        try
        {
            System.out.println("Listening mode...");
            while(true)
            {
                BufferedReader in=new BufferedReader(new
                InputStreamReader(System.in));
                s1=ss1.accept();
                s2=ss2.accept();
                sfcl=new BufferedReader(new
                InputStreamReader(s1.getInputStream()));
                sfc2=new BufferedReader(new
                InputStreamReader(s2.getInputStream()));
                System.out.println(sfcl.readLine());
                System.out.println(sfc2.readLine());
                S2c1=new DataOutputStream(s1.getOutputStream());
                S2c2=new DataOutputStream(s2.getOutputStream());
                System.out.println("data to be sent to client 1");
                s11=in.readLine();
                System.out.println("data to be sent to client 2");
                s12=in.readLine();
                S2c1.writeBytes(s11+'\n');
                S2c2.writeBytes(s12+'\n');
```

```

    }
    }
    catch(Exception e)
    {
    System.out.print(e);
    }
    finally
    {
    try
    {
    s1.close();
    s2.close();
    }
    catch(Exception e)
    {
    System.out.print(e);
    }
    }
    }
    public static void main(String args[])throws
    IOException,UnknownHostException
    {
    new multiserver1();
    }
    }

```

CLIENT SIDE PROGRAM

```

import java.io.*;
import java.net.*;
public class multiClient1
{
DataOutputStream StreamToServer;
BufferedReader StreamFromServer;
public multiClient1()
{
while(true)
{
try
{
String s;
Socket toServer=new Socket("10.10.0.114",1001);
StreamToServer=new
DataOutputStream(toServer.getOutputStream());
System.out.println("\n:1:");
BufferedReader r=new BufferedReader(new
InputStreamReader(System.in));
s=r.readLine();
StreamToServer.writeBytes(s+"\n");
StreamFromServer=new BufferedReader(new
InputStreamReader(toServer.getInputStream()));
String str=StreamFromServer.readLine();
System.out.println("\n:2:"+str);
}
catch(Exception e)
{
System.out.println(e);
}
}
}
}

```

```
public static void main(String args[])throws
IOException,UnknownHostException
{
new multiClient1();
}
}
```

CLIENT SIDE PROGRAM 2

```
import java.io.*;
import java.net.*;
public class multiClient2
{
DataOutputStream StreamToServer;
BufferedReader StreamFromServer;
public multiClient2()
{
while(true)
{
try
{
String s;
Socket toServer=new Socket("10.10.0.114",1002);
StreamToServer=new
DataOutputStream(toServer.getOutputStream());
System.out.println("\n:1:");
BufferedReader r=new BufferedReader(new
InputStreamReader(System.in));
s=r.readLine();
StreamToServer.writeBytes(s+"\n");
StreamFromServer=new BufferedReader(new
InputStreamReader(toServer.getInputStream()));
String str=StreamFromServer.readLine();
System.out.println("\n:2:"+str);
}
catch(Exception e)
{
System.out.println(e);
}
}
}
public static void main(String args[])throws
IOException,UnknownHostException
{
new multiClient2();
}
}
```


Q 7: Write a program to create a chat server that connects two clients that communicate with server using different ports.

[SCREENSHOTS]

Client Side 1:

```
D:\Java\jdk1.6.0_30\bin\ONETWORKING>
java multiClient2

:1:
Neil here

:2: hi! Mathew

:1:
```

Client Side 2:

```
D:\Java\jdk1.6.0_30\bin\ONETWORKING>
java multiClient1

:1:
Mathew here

:2: hi! Neil

:1:
```

Server Side:

```
D:\Java\jdk1.6.0_30\bin\ONETWORKING>java
multiserver1

Listening mode...
Neil here
Mathew here
data to be sent to client 1
hi! Neil
data to be sent to client 2
hi! Mathew
```

Q 8: Write a program to create a client server application to send the encrypted data from the client and then decrypting it at the server end.

[SOURCE CODE]

SERVER SIDE PROGRAM

```
import java.io.*;
import java.net.*;
public class server
{
public static void main(String args[])throws
IOException,UnknownHostException
{
String s3="",s2;
char c;
int i;
ServerSocket SS=new ServerSocket(1001);
System.out.println("\nListening Mode....");
while(true)
{
Socket S=SS.accept();
BufferedReader bin=new BufferedReader(new
InputStreamReader(S.getInputStream()));
DataOutputStream dl=new DataOutputStream(S.getOutputStream());
s2=bin.readLine();
System.out.println("Encrypted String is:" +s2);
for(i=0;i<s2.length();i++)
{
c=(char) (s2.charAt(i)-3);
s3=s3+c;
}
System.out.println("Decrypted string is:"+ s3);
}
}
}
```

CLIENT SIDE PROGRAM

```
import java.io.*;
import java.net.*;
class client
{
public static void main(String args[])throws
UnknownHostException,IOException
{
String s1="",sin;
char c;
int i;
System.out.println("Enter String ");
BufferedReader bclient=new BufferedReader(new
InputStreamReader(System.in));
sin=bclient.readLine();
for(i=0;i<sin.length();i++)
{
c=(char) (sin.charAt(i)+3);
s1=s1+c;
}
Socket s=new Socket("localhost",1001);
DataOutputStream dos=new DataOutputStream(s.getOutputStream());
dos.writeBytes(s1+'\n');
s.close();
}
}
```

Q 8: Write a program to create a client server application to send the encrypted data from the client and then decrypting it at the server end.

[SCREENSHOTS]

Client Side:

```
D:\Java\jdk1.6.0_30\bin\ONETWORKING>java client
Enter String
Neil
```

Server Side:

```
D:\Java\jdk1.6.0_30\bin\ONETWORKING>java server
Listening Mode...
Encrypted string is: ohlo
Decrypted string is: Neil
```

Q 9: Write a program to implement two way chatting between client and server.

[SOURCE CODE]

CLIENT SIDE PROGRAM

```
import java.io.*;
import java.net.*;
public class GossipClient
{
public static void main(String[] args) throws Exception
{
    Socket sock = new Socket("127.0.0.1", 3000);
    BufferedReader keyRead = new BufferedReader(new
    InputStreamReader(System.in));
    OutputStream ostream = sock.getOutputStream();
    PrintWriter pwrite = new PrintWriter(ostream, true);
        InputStream istream = sock.getInputStream();
    BufferedReader receiveRead = new BufferedReader(new
    InputStreamReader(istream));
        System.out.println("Start the chitchat, type
and press Enter key");
        String receiveMessage, sendMessage;
while(true)
{
    sendMessage = keyRead.readLine();    // keyboard reading
    pwrite.println(sendMessage);        // sending to server
    System.out.flush();                // flush the data
        if((receiveMessage =
receiveRead.readLine()) != null) //receive from server
    {
        System.out.println(receiveMessage); // displaying at DOS prompt
    }
}
    }
}
```

SERVER SIDE PROGRAM

```
import java.io.*;
import java.net.*;
public class GossipServer
{
    public static void main(String[] args) throws Exception
    {
        ServerSocket sersock = new ServerSocket(3000);
        System.out.println("Server ready for chatting");
        Socket sock = sersock.accept( );
                BufferedReader keyRead = new
        BufferedReader(new InputStreamReader(System.in));
        OutputStream ostream = sock.getOutputStream();
        PrintWriter pwrite = new PrintWriter(ostream, true);
                InputStream istream = sock.getInputStream();
        BufferedReader receiveRead = new BufferedReader(new
        InputStreamReader(istream));
                String receiveMessage, sendMessage;

        while(true)
        {
            if((receiveMessage = receiveRead.readLine( )) != null)
            {
                System.out.println(receiveMessage);
            }
            sendMessage = keyRead.readLine();
            pwrite.println(sendMessage);
            System.out.flush();
        }
    }
}
```

Q 9: Write a program to implement two way chatting between client and server.

[SCREENSHOTS]

Server Side:

```
D:\Java\jdk1.6.0_30\bin\ONETWORKING>
java GossipServer

Server ready for chatting
Neil here!
Hi! Neil
what are you doing?
Nothing...
```

Server Side:

```
D:\Java\jdk1.6.0_30\bin\ONETWORKING>
java GossipClient

Start the chitchat, type and press Enter key

Neil here!

Hi! Neil

what are you doing?
```

Q 10: Write a program for securely exchanging key between server and the client without sharing the private data.

[SOURCE CODE]

CLIENT SIDE PROGRAM

```
import java.net.*;
import java.io.*;
import java.math.*;
import java.lang.Integer;
public class KeyClient
{
private double a;
public double p,g;
double interResult()
{
return (Math.pow( g,a ))% p;
}
double result(double s)
{
return (Math.pow(s,a) )%p;
}
KeyClient(double l,double m, double n)
{
a=l;
p=m;
g=n;
}
public static void main(String args[]) throws Exception
{
KeyClient k=new KeyClient(6.0,23.0,5.0);
String s1="",s2="";
int ir=(int) (k.interResult()),fr;
System.out.println("Shared value = "+ir);
Socket s= new Socket("localhost",1400);
BufferedReader br= new BufferedReader(new
InputStreamReader(s.getInputStream()));
PrintStream dos=new PrintStream(s.getOutputStream());
System.out.println(s1);
dos.println(ir);
s2=br.readLine();
int r=Integer.parseInt(s2);
fr=(int) (k.result(r));
System.out.println("Final result = "+fr);
}
}
```


SERVER SIDE PROGRAM

```
import java.net.*;
import java.io.*;
import java.math.*;
import java.lang.Integer;
public class KeyServer
{
private int b;
public int pb,gb;
int interResult()
{
return ((int)Math.pow(gb,b))%pb;
}
int result(int s)
{
return ((int)Math.pow(s,b))%pb;
}
KeyServer(int l,int m, int n)
{
b=l;
pb=m;
gb=n;
}
public static void main(String args[]) throws Exception
{
KeyServer ks=new KeyServer(4,23,5);
String s1="",s2="";
int ir=(int)(ks.interResult());
int fr=0;
System.out.println("Shared value = "+ir);
ServerSocket ss= new ServerSocket(1400);
Socket s=ss.accept();
BufferedReader br= new BufferedReader(new
InputStreamReader(s.getInputStream()));
PrintStream dos=new PrintStream(s.getOutputStream());
System.out.println(s1);
dos.println(ir);
s2=br.readLine();
int r=Integer.parseInt(s2);
fr=(int)(ks.result(r));
System.out.println("Final result = "+fr);
s.close();
}
}
```

Q 10: Write a program for securely exchanging key between server and the client without sharing the private data.

[SCREENSHOTS]

Server Side:

```
D:\Java\jdk1.6.0_30\bin\ONETWORKING>
java KeyServer

Shared value = 4

Final result = 2
```

Client Side:

```
D:\Java\jdk1.6.0_30\bin\ONETWORKING>
java KeyClient

Shared value = 8

Final result = 2
```