Program: echo

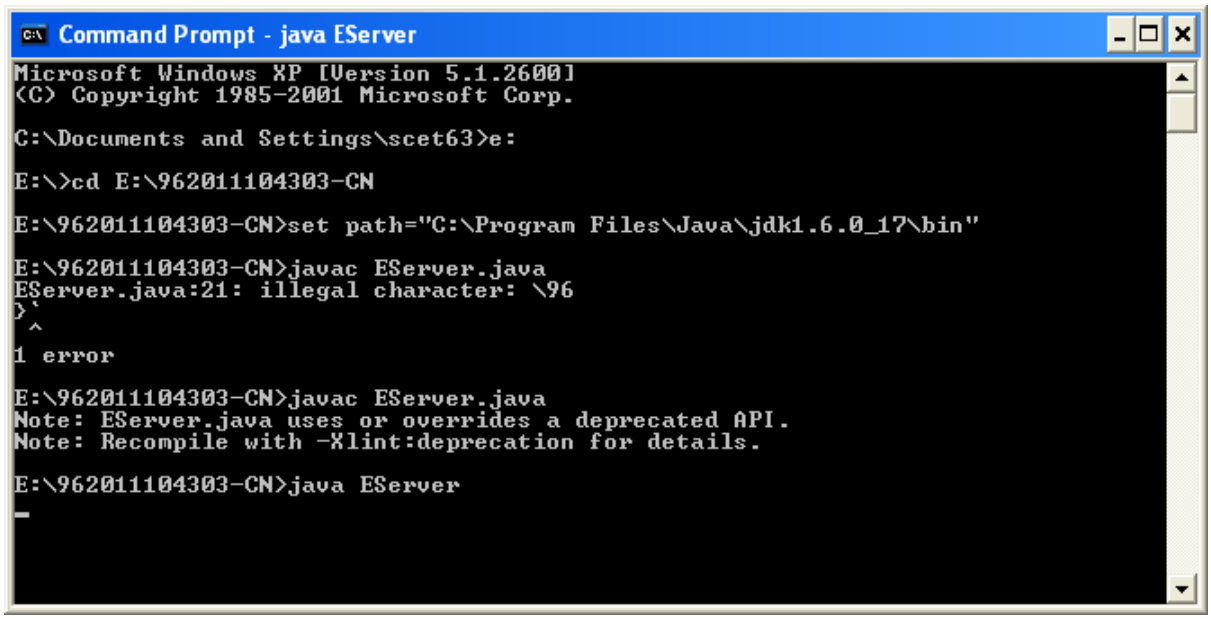
Eserver

```
import java.net.*;
import java.io.*;
public class EServer
{
public static void main(String args[])throws Exception
{
ServerSocket ss=new ServerSocket(8020);
String line;
DataInputStream dis;
PrintStream ps;
Socket s;
s=ss.accept();
dis=new DataInputStream(s.getInputStream());
ps=new PrintStream(s.getOutputStream());
while(true)
{
line=dis.readLine();
ps.println(line);
}
}
}
```

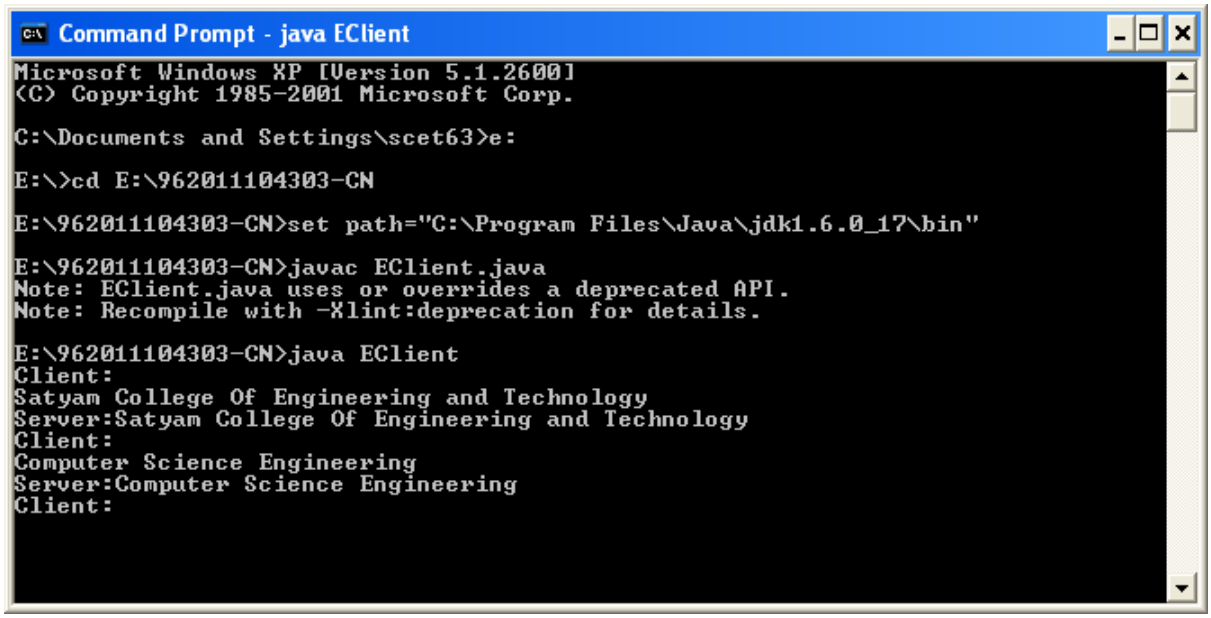
EClient

```
import java.net.*;
import java.io.*;
public class EClient
{
public static void main(String args[])throws Exception
{
InetAddress ia=InetAddress.getLocalHost();
Socket c=new Socket(ia,8020);
String line;
DataInputStream is,is1;
PrintStream os;
os=new PrintStream(c.getOutputStream());
is=new DataInputStream(System.in);
is1=new DataInputStream(c.getInputStream());
while(true)
{
System.out.println("Client:");
line=is.readLine();
os.println(line);
System.out.println("Server:"+is1.readLine());
}
}
}
```

EServer:



EClient:



Program: chat

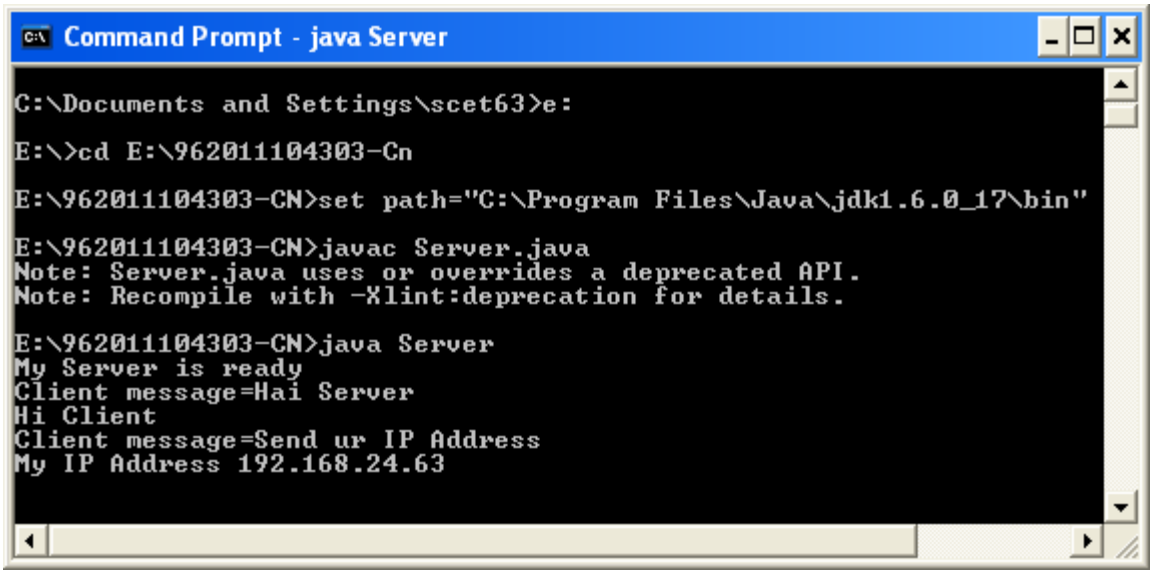
Server

```
import java.net.*;

import java.io.*;

public class Server

{

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

{

ServerSocket ss=new ServerSocket(3000);

Socket s=ss.accept();

System.out.println("My Server is Ready: ");

DataInputStream in=new DataInputStream(s.getInputStream());

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

while(true)

{

String st=in.readLine();

System.out.println(st);

st=new DataInputStream(System.in).readLine();

ps.println("Server message=  "+st);

}

}

}
```

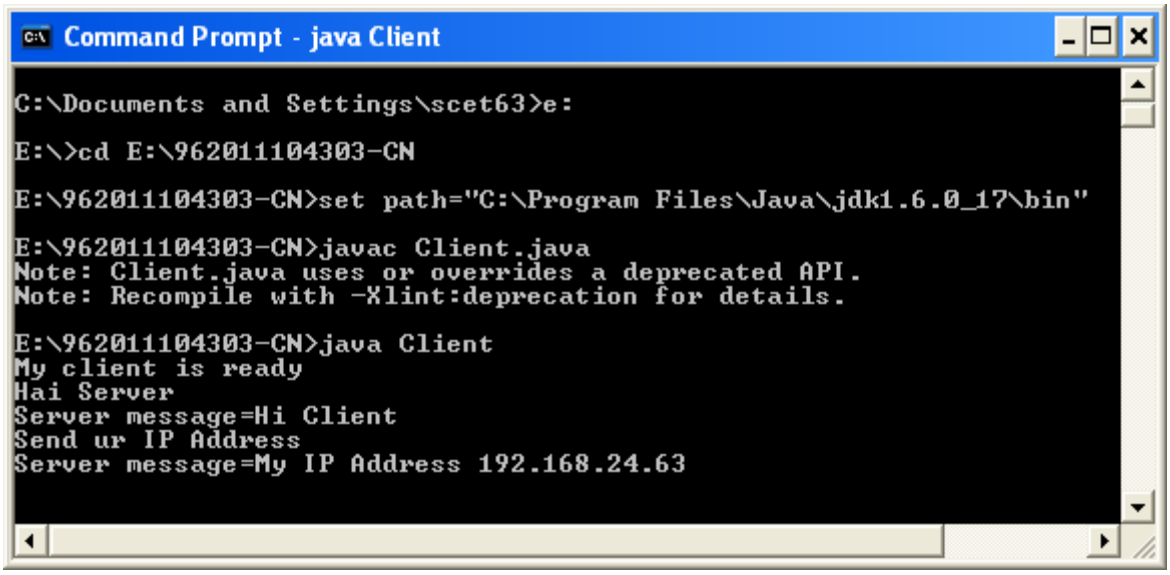
Client

```
import java.net.*;

import java.io.*;

public class Client

{

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

{

InetAddress ina=InetAddress.getLocalHost();

Socket s=new Socket(ina,3000);

System.out.println("My Client is Ready: ");

DataInputStream dis=new DataInputStream(s.getInputStream());

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

while(true)

{

String st=new DataInputStream(System.in).readLine();

ps.println("Client message=  "+st);

st=dis.readLine();

System.out.println(st);

}

}

}
```

Server:



Client:



Program:

//ARP SERVER

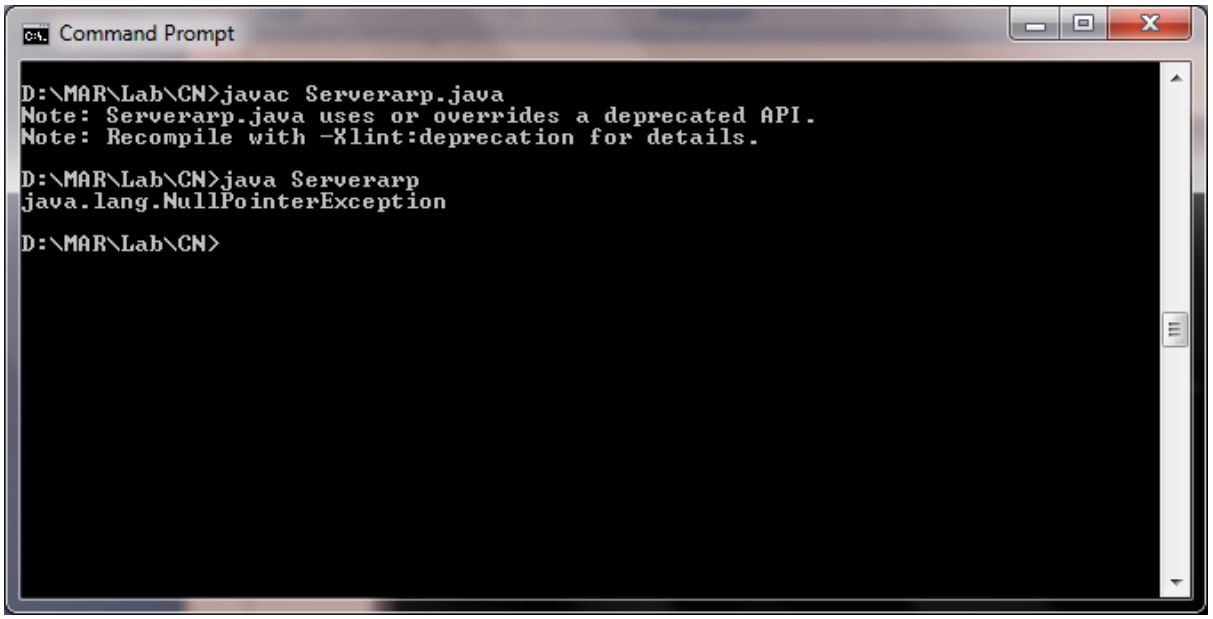
```
import java.io.*;
import java.net.*;
import java.util.*;
 class Serverarp
{
public static void main(String args[])
{
try
{
ServerSocket obj=new ServerSocket(139);
Socket obj1=obj.accept();
while(true)
{
DataInputStream din=new DataInputStream(obj1.getInputStream());
DataOutputStream dout=new DataOutputStream(obj1.getOutputStream());
 String str=din.readLine();
String ip[]={"165.165.80.80","165.165.79.1"};
String mac[]={"6A:08:AA:C2","8A:BC:E3:FA"};
for(int i=0;i<ip.length;i++)
{
if(str.equals(ip[i]))
{
```

```
dout.writeBytes(mac[i]+'\n');

break;

}

}

obj.close();

}

}

catch(Exception e)

{

System.out.println(e);

}

}

}
```

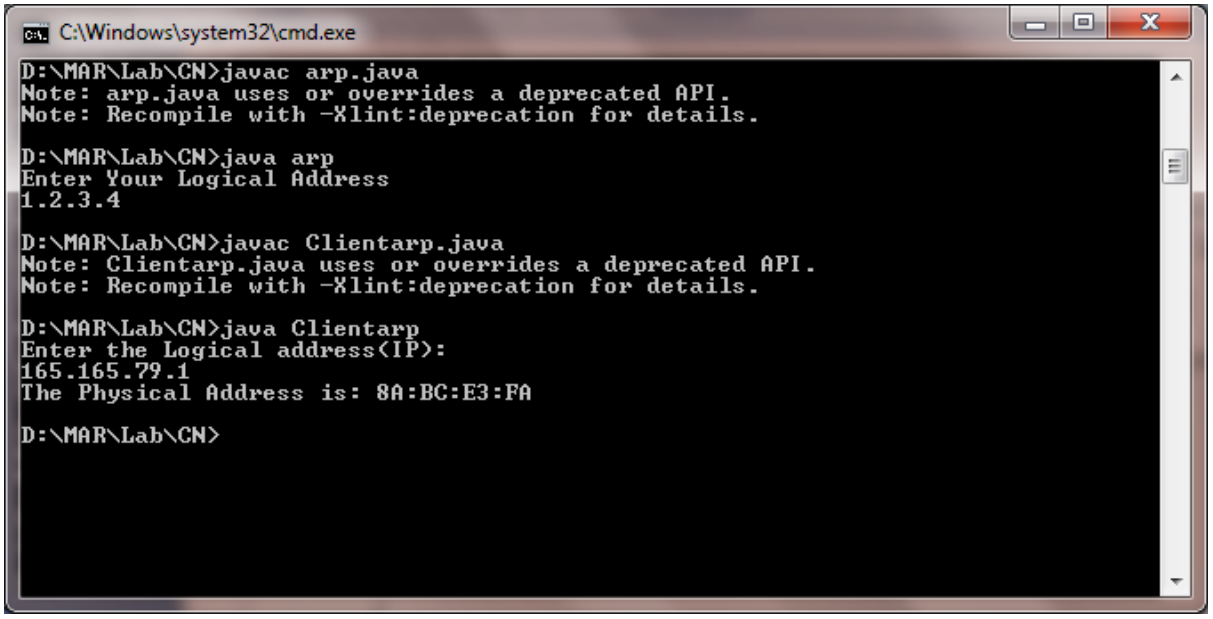
//ARP CLIENT

```
import java.io.*;
import java.net.*;
import java.util.*;
class Clientarp{
public static void main(String args[]){
try{
BufferedReader in=new BufferedReader(new InputStreamReader(System.in));
Socket clsct=new Socket("127.0.0.1",139);
DataInputStream din=new DataInputStream(clsct.getInputStream());
DataOutputStream dout=new DataOutputStream(clsct.getOutputStream());
System.out.println("Enter the Logical address(IP):");
String str1=in.readLine();
dout.writeBytes(str1+'\n');
String str=din.readLine();
System.out.println("The Physical Address is: "+str);
clsct.close();
}
catch (Exception e)
{
System.out.println(e);
}
}
}
```

Server:



Client:



Program: error

```
#include<stdio.h>

char m[50],g[50],r[50],q[50],temp[50];

void caltrans(int);

void crc(int);

void calram();

void shiftl();

int main()

{

int n,i=0;

char ch,flag=0;

printf("Enter the frame bits:");

while((ch=getc(stdin))!='\n')

m[i++]=ch;

n=i;

for(i=0;i<16;i++)

m[n++]='0';

m[n]='\0';

printf("Message after appending 16 zeros:%s",m);

for(i=0;i<=16;i++)

g[i]='0';

g[0]=g[4]=g[11]=g[16]='1';g[17]='\0';

printf("\ngenerator:%s\n",g);

crc(n);
```

```
printf("\n\nquotient:%s",q);
caltrans(n);
printf("\ntransmitted frame:%s",m);
printf("\nEnter transmitted frame:");
scanf("\n%s",m);
printf("CRC checking\n");
crc(n);
printf("\n\nlast remainder:%s",r);
for(i=0;i<16;i++)
if(r[i]!='0')
flag=1;
else
continue;
if(flag==1)
printf("Error during transmission");
else
printf("\n\nReceived freme is correct");
}
void crc(int n)
{
int i,j;
for(i=0;i<n;i++)
temp[i]=m[i];
for(i=0;i<16;i++)
```

```
r[i]=m[i];
printf("\nintermediate remainder\n");
for(i=0;i<n-16;i++)
{
if(r[0]=='1')
{
q[i]='1';
calram();
}
else
{
q[i]='0';
shiftl();
}
r[16]=m[17+i];
r[17]='\0';
printf("\nremainder %d:%s",i+1,r);
for(j=0;j<=17;j++)
temp[j]=r[j];
}
q[n-16]='\0';
}
void calram()
{
```

```
int i,j;
for(i=1;i<=16;i++)
r[i-1]=((int)temp[i]-48)^((int)g[i]-48)+48;
}
void shiftl()
{
int i;
for(i=1;i<=16;i++)
r[i-1]=r[i];
}
void caltrans(int n)
{
int i,k=0;
for(i=n-16;i<n;i++)
m[i]=((int)m[i]-48)^((int)r[k++]-48)+48;
m[i]='\0';
}
```

Output:

Enter the Frame Bits:

1011

The msg after appending 16 zeros:

10110000000000000000

The Transmitted frame is:10111011000101101011

Enter the transmitted Frame

10111011000101101011

Received msg:10111011000101101011

The Remainder is:0000000000000000

Received frame is correct.