Assignment No. 04

Go Back N (Java)

/*Server Program*/

```
import java.net.*;
import java.io.*;
import java.util.*;
public class Server
{
public static void main(String args[]) throws Exception
{
ServerSocket server=new ServerSocket(6262);
System.out.println("Server established.");
Socket client=server.accept();
ObjectOutputStream oos=new ObjectOutputStream(client.getOutputStream());
ObjectInputStream ois=new ObjectInputStream(client.getInputStream());
System.out.println("Client is now connected.");
int x=(Integer)ois.readObject();
int k=(Integer)ois.readObject();
int j=0;
int i=(Integer)ois.readObject();
boolean flag=true;
Random r=new Random(6);
int mod=r.nextInt(6);
while(mod==1||mod==0)
mod=r.nextInt(6);
while(true)
{
int c=k;
for(int h=0;h<=x;h++)
{
System.out.print("|"+c+"|");
c=(c+1)%x;
}
System.out.println();
System.out.println();
if(k==j)
{
System.out.println("Frame "+k+" recieved"+"\n"+"Data:"+j);
j++;
System.out.println();
}
else
System.out.println("Frames recieved not in correct order"+"\n"+" Expected
farme:" + j +"\n"+ " Recieved frame no :"+ k);
System.out.println();
if(j%mod==0 && flag)
{
System.out.println("Error found. Acknowledgement not sent. ");
flag=!flag;
j-;
}
else if(k==j-1)
{
oos.writeObject(k);
System.out.println("Acknowledgement sent");
}
System.out.println();
if(j%mod==0)
flag=!flag;
k=(Integer)ois.readObject();
```

```
if(k==-1)
break;
i=(Integer)ois.readObject();
}
System.out.println("Client finished sending data. Exiting");
oos.writeObject(-1);
}
}
/*Client Program*/
import java.util.*;
import java.net.*;
import java.io.*;
public class Client
{
public static void main(String args[]) throws Exception
{
BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
System.out.print("Enter the value of m : ");
int m=Integer.parseInt(br.readLine());
int x=(int)((Math.pow(2,m))-1);
System.out.print("Enter no. of frames to be sent:");
int count=Integer.parseInt(br.readLine());
int data[]=new int[count];
int h=0;
for(int i=0;i<count;i++)
{
System.out.print("Enter data for frame no " +h+ " => ");
data[i]=Integer.parseInt(br.readLine());
h=(h+1)%x;
}
Socket client=new Socket("localhost",6262);
ObjectInputStream ois=new ObjectInputStream(client.getInputStream());
ObjectOutputStream oos=new ObjectOutputStream(client.getOutputStream());
System.out.println("Connected with server.");
boolean flag=false;
GoBackNListener listener=new GoBackNListener(ois,x);
listener=new GoBackNListener(ois,x);
listener.t.start();
int strt=0;
h=0;
oos.writeObject(x);
do
{
int c=h;
for(int i=h;i<count;i++)
{
System.out.print("|"+c+"|");
c=(c+1)%x;
}
System.out.println();
System.out.println();
h=strt;
for(int i=strt;i<x;i++)
{
System.out.println("Sending frame:"+h);
h=(h+1)%x;
System.out.println();
oos.writeObject(i);
oos.writeObject(data[i]);
Thread.sleep(100);
}
```

```
listener.t.join(3500);
if(listener.reply!=x-1)
{
System.out.println("No reply from server in 3.5 seconds. Resending data
from frame no " + (listener.reply+1));
System.out.println();
strt=listener.reply+1;
flag=false;
}
else
{
System.out.println("All elements sent successfully. Exiting");
flag=true;
}
}while(!flag);
oos.writeObject(-1);
}
}
class GoBackNListener implements Runnable
{
Thread t;
ObjectInputStream ois;
int reply,x;
GoBackNListener(ObjectInputStream o,int i)
{
t=new Thread(this);
ois=o;
reply=-2;
x=i;
}
@Override
public void run() {
try
{
int temp=0;
while(reply!=-1)
{
reply=(Integer)ois.readObject();
if(reply!=-1 && reply!=temp+1)
reply=temp;
if(reply!=-1)
{
temp=reply;
System.out.println("Acknowledgement of frame no " + (reply%x) + "
recieved.");
System.out.println();
}
}
reply=temp;
}
catch(Exception e)
{
System.out.println("Exception => " + e);
}
}
}
/*Client Output
Enter the value of m : 7
Enter no. of frames to be sent:5
Enter data for frame no 0 => 1
```

```
Enter data for frame no 1 => 2
Enter data for frame no 2 => 3
Enter data for frame no 3 => 4
Enter data for frame no 4 => 5
Connected with server.
|0||1||2||3||4|
Sending frame:0
Acknowledgement of frame no 0 recieved.
Sending frame:1
Sending frame:2
Sending frame:3
Sending frame:4
Sending frame:5
*/
/*Server Output
Server established.
Client is now connected.
|0||1||2||3||4||5||6||7||8||9||10||11||12||13||14||15||16||17||18||19||20||
21||22||23||24||25||26||27||28||29||30||31||32||33||34||35||36||37||38||39|
|40||41||42||43||44||45||46||47||48||49||50||51||52||53||54||55||56||57||58
||59||60||61||62||63||64||65||66||67||68||69||70||71||72||73||74||75||76||7
7||78||79||80||81||82||83||84||85||86||87||88||89||90||91||92||93||94||95||
96||97||98||99||100||101||102||103||104||105||106||107||108||109||110||111|
|112||113||114||115||116||117||118||119||120||121||122||123||124||125||126|
|0|
Frame 0 recieved
Data:0
Acknowledgement sent
|1||2||3||4||5||6||7||8||9||10||11||12||13||14||15||16||17||18||19||20||21|
|22||23||24||25||26||27||28||29||30||31||32||33||34||35||36||37||38||39||40
||41||42||43||44||45||46||47||48||49||50||51||52||53||54||55||56||57||58||5
9||60||61||62||63||64||65||66||67||68||69||70||71||72||73||74||75||76||77||
78||79||80||81||82||83||84||85||86||87||88||89||90||91||92||93||94||95||96|
|97||98||99||100||101||102||103||104||105||106||107||108||109||110||111||11
2||113||114||115||116||117||118||119||120||121||122||123||124||125||126||0|
|1|
Frame 1 recieved
Data:1
Error found. Acknowledgement not sent.
|2||3||4||5||6||7||8||9||10||11||12||13||14||15||16||17||18||19||20||21||22
||23||24||25||26||27||28||29||30||31||32||33||34||35||36||37||38||39||40||4
1||42||43||44||45||46||47||48||49||50||51||52||53||54||55||56||57||58||59||
60||61||62||63||64||65||66||67||68||69||70||71||72||73||74||75||76||77||78|
|79||80||81||82||83||84||85||86||87||88||89||90||91||92||93||94||95||96||97
||98||99||100||101||102||103||104||105||106||107||108||109||110||111||112||
113||114||115||116||117||118||119||120||121||122||123||124||125||126||0||1|
|2|
Frames recieved not in correct order
Expected farme:1
Recieved frame no :2
|3||4||5||6||7||8||9||10||11||12||13||14||15||16||17||18||19||20||21||22||2
3||24||25||26||27||28||29||30||31||32||33||34||35||36||37||38||39||40||41||
42||43||44||45||46||47||48||49||50||51||52||53||54||55||56||57||58||59||60|
|61||62||63||64||65||66||67||68||69||70||71||72||73||74||75||76||77||78||79
||80||81||82||83||84||85||86||87||88||89||90||91||92||93||94||95||96||97||9
8||99||100||101||102||103||104||105||106||107||108||109||110||111||112||113
||114||115||116||117||118||119||120||121||122||123||124||125||126||0||1||2|
|3|
```

```
Frames recieved not in correct order
Expected farme:1
Recieved frame no :3
|4||5||6||7||8||9||10||11||12||13||14||15||16||17||18||19||20||21||22||23||
24||25||26||27||28||29||30||31||32||33||34||35||36||37||38||39||40||41||42|
|43||44||45||46||47||48||49||50||51||52||53||54||55||56||57||58||59||60||61
||62||63||64||65||66||67||68||69||70||71||72||73||74||75||76||77||78||79||8
0||81||82||83||84||85||86||87||88||89||90||91||92||93||94||95||96||97||98||
99||100||101||102||103||104||105||106||107||108||109||110||111||112||113||1
14||115||116||117||118||119||120||121||122||123||124||125||126||0||1||2||3|
|4|
Frames recieved not in correct order
Expected farme:1
Recieved frame no :4
*/
```

Selective Repeat

//SENDER SIDE

```
#include<stdio.h>
#include<sys/types.h>
#include<sys/socket.h>
#include<netinet/in.h>
#include<string.h>
#include<time.h>
#include<stdlib.h>
#include<ctype.h>
#define W 5

char a[10];
char b[10];
void alpha9(int);
int con();

int main()
{
    int s,f,wl,c=1,x,i=0,j,n,p=0,e=0;
    struct sockaddr_in ser;
    s=socket(AF_INET,SOCK_STREAM,0);
    ser.sin_family=AF_INET;
    ser.sin_port=6500;
    ser.sin_addr.s_addr=inet_addr("192.168.0.109");
    connect(s,(struct sockaddr *) &ser, sizeof(ser));
    printf("\nTCP Connection Established.\n");
    printf("\nEnter the number of Frames: ");
    scanf("%d",&f);
    alpha9(f);
    send(s,a,sizeof(a),0);
    strcpy(b,"Time Out ");
    while(1)
    {
        for(i=0;i<W;i++)
```

```
{
    alpha9(c);
    send(s,a,sizeof(a),0);
    if(c<=f)
    {
        printf("\nFrame %d Sent",c);
        c++;
    }
}
i=0;
wl=W;
while(i<W)
{
    recv(s,a,sizeof(a),0);
    p=atoi(a);
    if(a[0]=='N')
    {
        e=con();
        if(e<f)
        {
            printf("\nNAK %d",e);
            printf("\nFrame %d sent",e);
            i--;
        }
    }
    else
    {
        if(p<=f)
        {
            printf("\nFrame %s Acknowledged",a);
            wl--;
        }
        else
        {
            break;
        }
    }
    if(p>f)
    {
        break;
    }
    i++;
}
if(wl==0 && c>f)
{
    send(s,b,sizeof(b),0);
    break;
}
else
{
    c=c-wl;
```

```
            wl=W;
        }
    }
    close(s);
    return 0;
}

void alpha9(int z)
{
    int k,i=0,j,g;
    k=z;
    while(k>0)
    {
        i++;
        k=k/10;
    }
    g=i;
    i--;
    while(z>0)
    {
        k=z%10;
        a[i]=k+48;
        i--;
        z=z/10;
    }
    a[g]='\0';
}
int con()
{
    char k[9];
    int i=1;
    while(a[i]!='\0')
    {
        k[i-1]=a[i];
        i++;
    }
    k[i-1]='\0';
    i=atoi(k);
    return i;
}
```

*****OUTPUT*****

```
Aj:~$ cc SelRepS.c -o sender
Aj:~$ ./sender

TCP Connection Established.

Enter the number of Frames: 25

Frame 1 Sent
```

```
Frame 2 Sent
Frame 3 Sent
Frame 4 Sent
Frame 5 Sent
Frame 1 Acknowledged
Frame 2 Acknowledged
Frame 3 Acknowledged
NAK 4
Frame 4 sent
Frame 4 Acknowledged
Frame 5 Acknowledged
Frame 6 Sent
Frame 7 Sent
Frame 8 Sent
Frame 9 Sent
Frame 10 Sent
Frame 6 Acknowledged
Frame 7 Acknowledged
Frame 8 Acknowledged
Frame 9 Acknowledged
Frame 10 Acknowledged
Frame 11 Sent
Frame 12 Sent
Frame 13 Sent
Frame 14 Sent
Frame 15 Sent
NAK 11
Frame 11 sent
Frame 11 Acknowledged
NAK 12
Frame 12 sent
Frame 12 Acknowledged
Frame 13 Acknowledged
Frame 14 Acknowledged
Frame 15 Acknowledged
Frame 16 Sent
Frame 17 Sent
Frame 18 Sent
Frame 19 Sent
Frame 20 Sent
NAK 16
Frame 16 sent
Frame 16 Acknowledged
Frame 17 Acknowledged
NAK 18
Frame 18 sent
Frame 18 Acknowledged
NAK 19
Frame 19 sent
Frame 19 Acknowledged
Frame 20 Acknowledged
```

```
Frame 21 Sent
Frame 22 Sent
Frame 23 Sent
Frame 24 Sent
Frame 25 Sent
NAK 21
Frame 21 sent
Frame 21 Acknowledged
NAK 22
Frame 22 sent
Frame 22 Acknowledged
Frame 23 Acknowledged
Frame 24 Acknowledged
Frame 25 Sent
Aj:~$
```

//RECEIVER SIDE

```
#include<stdio.h>
#include<sys/types.h>
#include<sys/socket.h>
#include<netinet/in.h>
#include<string.h>
#include<time.h>
#include<stdlib.h>
#include<ctype.h>
#include<arpa/inet.h>

#define W 5
#define P1 50
#define P2 10

char a[10];
char b[10];
void alpha9(int);
void alp(int);

int main()
{
    struct sockaddr_in ser,cli;
    int s,n,sock,i,j,c=1,f;
    unsigned int s1;
    s=socket(AF_INET,SOCK_STREAM,0);
    ser.sin_family=AF_INET;
    ser.sin_port=6500;
    ser.sin_addr.s_addr=inet_addr("192.168.0.109");
    bind(s,(struct sockaddr *) &ser, sizeof(ser));
    listen(s,1);
    n=sizeof(cli);
    sock=accept(s,(struct sockaddr *)&cli, &n);
    printf("\nTCP Connection Established.\n");
```

```
    s1=(unsigned int) time(NULL);
    srand(s1);
    strcpy(b,"Time Out ");
    recv(sock,a,sizeof(a),0);
    f=atoi(a);
    while(1)
    {
        for(i=0;i<W;i++)
        {
            recv(sock,a,sizeof(a),0);
            if(strcmp(a,b)==0)
            {
                break;
            }
        }
        i=0;
        while(i<W)
        {
            L:
                j=rand()%P1;
            if(j<P2)
            {
                alp(c);
                send(sock,b,sizeof(b),0);
                goto L;
            }
            else
            {
                alpha9(c);
                if(c<=f)
                {
                    printf("\nFrame %s Received ",a);
                    send(sock,a,sizeof(a),0);
                }
                else
                {
                    break;
                }
                c++;
            }
            if(c>f)
            {
                break;
            }
            i++;
        }
    }
    close(sock);
    close(s);
    return 0;
}
```

```
void alpha9(int z)
{
    int k,i=0,j,g;
    k=z;
    while(k>0)
    {
        i++;
        k=k/10;
    }
    g=i;
    i--;
    while(z>0)
    {
        k=z%10;
        a[i]=k+48;
        i--;
        z=z/10;
    }
    a[g]='\0';
}

void alp(int z)
{
    int k,i=1,j,g;
    k=z;
    b[0]='N';
    while(k>0)
    {
        i++;
        k=k/10;
    }
    g=i;
    i--;
    while(z>0)
    {
        k=z%10;
        b[i]=k+48;
        i--;
        z=z/10;
    }
    b[g]='\0';
}
```

*****OUTPUT*****

```
Aj:~$ cc SelRepR.c -o receiver
Aj:~$ ./receiver

TCP Connection Established.
```

```
Frame 1 Received
Frame 2 Received
Frame 3 Received
Frame 4 Received
Frame 5 Received
Frame 6 Received
Frame 7 Received
Frame 8 Received
Frame 9 Received
Frame 10 Received
Frame 11 Received
Frame 12 Received
Frame 13 Received
Frame 14 Received
Frame 15 Received
Frame 16 Received
Frame 17 Received
Frame 18 Received
Frame 19 Received
Frame 20 Received
Frame 21 Received
Frame 22 Received
Frame 23 Received
Frame 24 Received
Aj:~$
```

Assignment No. 05 (Subnetting)

```
import java.io.*;

class Ipfind
{
                public static void main(String[] args)throws IOException {

                                DataInputStream dis = new
DataInputStream(System.in);
                                System.out.println("Enter IP Address (eg:
192.168.1.1)");
                                String ipAddr = dis.readLine();
                                String[] ipAddrParts=ipAddr.split("\\.");
                                String mask="";

                                int firstoctet =
Integer.parseInt(ipAddrParts[0]);
                                if(firstoctet<=127)
                                {
                                                mask = "255.0.0.0";
                                                System.out.println("Class A
IP Address");
                                                System.out.println("The
Subnet mask is: "+mask);
                                }
                                else if(firstoctet>=128 && firstoctet<=191)
                                {
                                                mask = "255.255.0.0";
                                                System.out.println("Class B
IP Address");
                                                System.out.println("The
Subnet mask is: "+mask);
                                }
                                else if(firstoctet>=192 && firstoctet<=223)
                                {
                                                mask = "255.255.255.0";
                                                System.out.println("Class C
IP Address");
                                                System.out.println("The
Subnet mask is: "+mask);
                                }
                                else if(firstoctet>=224 && firstoctet<=239)
                                {
                                                mask = "255.0.0.0";
                                                System.out.println("Class D
IP Address; Used for multicasting");
                                }
                                else if(firstoctet>=240 && firstoctet<=254)
                                {
                                                mask = "255.0.0.0";
                                                System.out.println("Class D
IP Address; Experimental Use");
```

```
                                }



                                String[] maskParts=mask.split("\\.");
                                String firstAddr="";
                                String lastAddr="";

                                for(int i=0;i<4;i++){
                int x=Integer.parseInt(ipAddrParts[i]);
                int y=Integer.parseInt(maskParts[i]);
                int z=x&y;
                int w=z|(y^255);               //last ip = ipaddress &&
subnetmask + ~subnetmask
                firstAddr+=z+".";
                lastAddr+=w+".";
            }

        System.out.println("First IP of block: "+firstAddr);
        System.out.println("Last IP of block: "+lastAddr);

                }
}
/*output:
bappi@bappi-Inspiron-3543:~$ java Ipfind
Enter IP Address (eg: 192.168.1.1)
192.168.0.1
Class C IP Address
The Subnet mask is: 255.255.255.0
First IP of block: 192.168.0.0.
Last IP of block: 192.168.0.255.
bappi@bappi-Inspiron-3543:~$
*/
```

Assignment No. 06

(Message Transfer)

//CLIENT SIDE

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <netdb.h>
#include<arpa/inet.h>
#include<unistd.h>

void error(const char *msg)
{
    perror(msg);
    exit(0);
}

int main(int argc, char *argv[])
{
    int sockfd, portno, n;
    struct sockaddr_in serv_addr;
    struct hostent *server; char fname[25];

    char buffer[256];
    if (argc < 3) {
       fprintf(stderr,"usage %s hostname port\n", argv[0]);
       exit(0);
    }
    portno = atoi(argv[2]);
    sockfd = socket(AF_INET, SOCK_STREAM, 0);
    if (sockfd < 0)
        error("ERROR opening socket");
    server = gethostbyname(argv[1]);
    if (server == NULL) {
        fprintf(stderr,"ERROR, no such host\n");
        exit(0);
    }
    bzero((char *) &serv_addr, sizeof(serv_addr));
    serv_addr.sin_family = AF_INET;
    bcopy((char *)server->h_addr,
         (char *)&serv_addr.sin_addr.s_addr,
         server->h_length);
    serv_addr.sin_port = htons(portno);
    if (connect(sockfd,(struct sockaddr *) &serv_addr,sizeof(serv_addr)) <
0)
        error("ERROR connecting");
```

```
    bzero(buffer,256);

    printf(" Enter the message: ");
    bzero(buffer,256);
    fgets(buffer,255,stdin);
    n = write(sockfd,buffer,strlen(buffer));
    if (n < 0)
         error("ERROR writing to socket");
    bzero(buffer,256);
    n = read(sockfd,buffer,255);
    if (n < 0)
         error("ERROR reading from socket");
    printf("%s\n",buffer);
close(sockfd);
return 0;
}
```

*****OUTPUT*****

```
gaurav:~$ cc client2.c -o c2
gaurav:~$ ./c2 192.168.0.109 6000
 Enter the message: Hello Server
I got your message
gaurav:~$
```

//SERVER SIDE

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include<arpa/inet.h>

void error(const char *msg)
{
    perror(msg);
    exit(1);
}

int main(int argc, char *argv[])
{
     int sockfd, newsockfd, portno;
     socklen_t clilen;
     char buffer[256];
     struct sockaddr_in serv_addr, cli_addr;
     int n;
     if (argc < 2) {
```

```
         fprintf(stderr,"ERROR, no port provided\n");
         exit(1);
     }
     sockfd = socket(AF_INET, SOCK_STREAM, 0);
     if (sockfd < 0)
        error("ERROR opening socket");
     bzero((char *) &serv_addr, sizeof(serv_addr));
     portno = atoi(argv[1]);
     serv_addr.sin_family = AF_INET;
     serv_addr.sin_addr.s_addr = INADDR_ANY;
     serv_addr.sin_port = htons(portno);
     if (bind(sockfd, (struct sockaddr *) &serv_addr,
              sizeof(serv_addr)) < 0)
              error("ERROR on binding");
     listen(sockfd,5);
     clilen = sizeof(cli_addr);
     newsockfd = accept(sockfd,
                 (struct sockaddr *) &cli_addr,
                 &clilen);
     if (newsockfd < 0)
          error("ERROR on accept");
     bzero(buffer,256);
    n = read(newsockfd,buffer,255);
     if (n < 0) error("ERROR reading from socket");
     printf("Here is the message: %s\n",buffer);
     n = write(newsockfd,"I got your message",18);
     if (n < 0) error("ERROR writing to socket");

close(newsockfd);
close(sockfd);
return 0;
}
```

*****OUTPUT*****

```
gaurav:~$ cc server2.c -o s2
gaurav:~$ ./s2 6000
Here is the message: Hello Server
```

(File Transfer)
TCP File transfer

//CLIENT SIDE

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
#include <sys/types.h>
#include <sys/socket.h>
```

```
#include <netinet/in.h>
#include <netdb.h>
#include<arpa/inet.h>
#include<unistd.h>
#define PORT 6500

void error(const char *msg)
{
    perror(msg);
    exit(0);
}

int main(int argc, char *argv[])
{
    int sockfd, portno, n;
    struct sockaddr_in serv_addr;
    struct hostent *server; char fname[25];

    char buffer[1000];
    if (argc < 3) {
       fprintf(stderr,"usage %s hostname port\n", argv[0]);
       exit(0);
    }
    portno = PORT;
    sockfd = socket(AF_INET, SOCK_STREAM, 0);
    if (sockfd < 0)
        error("ERROR opening socket");
    server = gethostbyname(argv[1]);
    if (server == NULL) {
        fprintf(stderr,"ERROR, no such host\n");
        exit(0);
    }
    bzero((char *) &serv_addr, sizeof(serv_addr));
    serv_addr.sin_family = AF_INET;
    bcopy((char *)server->h_addr,
         (char *)&serv_addr.sin_addr.s_addr,
         server->h_length);
    serv_addr.sin_port = htons(portno);
    if (connect(sockfd,(struct sockaddr *) &serv_addr,sizeof(serv_addr)) <
0)
        error("ERROR connecting");
    bzero(buffer,1000);

FILE *f;
write(sockfd,argv[2],sizeof(argv[2]));
f=fopen(argv[2],"r");
fread(buffer,1000,1,f);
write(sockfd,buffer,1000);
bzero(buffer,1000);
read(sockfd,buffer,1000);
printf("%s\n",buffer);
```

```
close(sockfd);
return 0;
}
```

*****OUTPUT*****

```
Aj:~$ cc client1.c -o c1
Aj:~$ ./c1 192.168.0.109 test.txt
The file has been sent successfully
Aj:~$
```

//SERVER SIDE

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include<arpa/inet.h>
#define PORT 6500

void error(const char *msg)
{
    perror(msg);
    exit(1);
}

int main(int argc, char *argv[])
{
     int sockfd, newsockfd, portno;
     socklen_t clilen;
     char buffer[1000];char fname[25];
     struct sockaddr_in serv_addr, cli_addr;
     int n;
     if (argc < 1) {
         fprintf(stderr,"ERROR, no port provided\n");
         exit(1);
     }
     sockfd = socket(AF_INET, SOCK_STREAM, 0);
     if (sockfd < 0)
        error("ERROR opening socket");
     bzero((char *) &serv_addr, sizeof(serv_addr));
     portno = PORT;
     serv_addr.sin_family = AF_INET;
     serv_addr.sin_addr.s_addr = INADDR_ANY;
     serv_addr.sin_port = htons(portno);
     if (bind(sockfd, (struct sockaddr *) &serv_addr,
              sizeof(serv_addr)) < 0)
              error("ERROR on binding");
```

```
     listen(sockfd,5);
     clilen = sizeof(cli_addr);
     newsockfd = accept(sockfd,
                 (struct sockaddr *) &cli_addr,
                 &clilen);
     if (newsockfd < 0)
          error("ERROR on accept");
     bzero(buffer,1000);

FILE *fp;
read(newsockfd,fname,sizeof(fname));
fp=fopen(fname,"w");
read(newsockfd,buffer,1000);
fwrite(buffer,strlen(buffer),1,fp);
write(newsockfd,"The file has been sent successfully",30);
printf("The contents of the file are  : \n\n");
printf("%s\n",buffer);
close(fp);
close(newsockfd);
close(sockfd);
return 0;
}
```

*****OUTPUT*****

```
Aj:~$ ./s1
The contents of the file are  :

MS SQL Server
MS SQL Server is a Relational Database Management System developed by
Microsoft Inc.
Its primary query languages are:
T-SQL
ANSI SQL

Aj:~$
*/
```

(Arithmetic Calculator)
Arithmatic calculator:

//CLIENT SIDE

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <netdb.h>
#include<arpa/inet.h>
```

```
#include<unistd.h>


void error(const char *msg)
{
    perror(msg);
    exit(0);
}

int main(int argc, char *argv[])
{
    int sockfd, portno, n;
    struct sockaddr_in serv_addr;
    struct hostent *server; char fname[25];
    int n1,n2,ans,choice,yes;
    char buffer[256];
    char s_num[5];
      int num;
    if (argc < 3) {
       fprintf(stderr,"usage %s hostname port\n", argv[0]);
       exit(0);
    }
    portno = atoi(argv[2]);
    sockfd = socket(AF_INET, SOCK_STREAM, 0);
    if (sockfd < 0)
        error("ERROR opening socket");
    server = gethostbyname(argv[1]);
    if (server == NULL) {
        fprintf(stderr,"ERROR, no such host\n");
        exit(0);
    }
    bzero((char *) &serv_addr, sizeof(serv_addr));
    serv_addr.sin_family = AF_INET;
    bcopy((char *)server->h_addr,
         (char *)&serv_addr.sin_addr.s_addr,
         server->h_length);
    serv_addr.sin_port = htons(portno);
    if (connect(sockfd,(struct sockaddr *) &serv_addr,sizeof(serv_addr)) <
0)
        error("ERROR connecting");
A : bzero(buffer,256);
read(sockfd,buffer,256);
printf("%s\n",buffer);
bzero(buffer,256);
scanf("%d",&choice);
write(sockfd,&choice,sizeof(int));
if(choice==5)
goto T;
read(sockfd,buffer,256);
printf("%s\n",buffer);
bzero(buffer,256);
```

```
scanf("%d",&n1);
write(sockfd,&n1,sizeof(int));
read(sockfd,buffer,256);
printf("%s\n",buffer);
bzero(buffer,256);
scanf("%d",&n2);
write(sockfd,&n2,sizeof(int));
read(sockfd,&ans,sizeof(int));
printf("Server-   The Answer is : %d\n",ans);
read(sockfd,buffer,256);
printf("%s\n",buffer);
scanf("%d",&yes);
write(sockfd,&yes,sizeof(int));
if(yes==1)
goto A;

T :   bzero(buffer,256);
      read(sockfd,buffer,256);
      printf("%s",buffer);
      close(sockfd);
return 0;
}
```

*****OUTPUT*****

```
Aj:~$ cc client3.c -o c3
Aj:~$ ./c3 192.168.0.109 60005
Server-     *******CALCULATOR*******
1. ADDITION
2. SUBTRACTION
3. MULTIPLICATION
4. DIVISION
5. EXIT

ENTER  YOUR  CHOICE
1
Server-   Enter the First Number
4
Server-   Enter the Second Number
5
Server-   The Answer is : 9
Server-   Do You Want More Arithmetic(1/0)  ?
1
Server-     *******CALCULATOR*******
1. ADDITION
2. SUBTRACTION
3. MULTIPLICATION
4. DIVISION
5. EXIT

ENTER  YOUR  CHOICE
```

```
3
Server-   Enter the First Number
6
Server-   Enter the Second Number
7
Server-   The Answer is : 42
Server-   Do You Want More Arithmetic(1/0)  ?
0
Server-   Exited
Aj:~$
```

//SERVER SIDE

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include<arpa/inet.h>

void error(const char *msg)
{
    perror(msg);
```

```
    exit(1);
}

int main(int argc, char *argv[])
{
     int sockfd, newsockfd, portno;
     socklen_t clilen;
     char menu[100]="NO VALUE";
     char buffer[256],m[5];
     struct sockaddr_in serv_addr, cli_addr;
     int n,n1,n2,ans,choice,yes;
     if (argc < 2) {
         fprintf(stderr,"ERROR, no port provided\n");
         exit(1);
     }
     sockfd = socket(AF_INET, SOCK_STREAM, 0);
     if (sockfd < 0)
        error("ERROR opening socket");
     bzero((char *) &serv_addr, sizeof(serv_addr));
     portno = atoi(argv[1]);
     serv_addr.sin_family = AF_INET;
     serv_addr.sin_addr.s_addr = INADDR_ANY;
     serv_addr.sin_port = htons(portno);
     if (bind(sockfd, (struct sockaddr *) &serv_addr,
              sizeof(serv_addr)) < 0)
              error("ERROR on binding");
     listen(sockfd,5);
     clilen = sizeof(cli_addr);
     newsockfd = accept(sockfd,
                 (struct sockaddr *) &cli_addr,
                 &clilen);
     if (newsockfd < 0)
          error("ERROR on accept");
     bzero(buffer,256);

S : strcpy(menu,"Server-     *******CALCULATOR*******\n1. ADDITION\n2.
SUBTRACTION\n3. MULTIPLICATION\n4. DIVISION\n5. EXIT\n\nENTER  YOUR
CHOICE");
write(newsockfd,menu,strlen(menu));
read(newsockfd,&choice,sizeof(int));
printf("Client-   The choice is : %d\n",choice);
if(choice==5)
goto M;
write(newsockfd,"Server-   Enter the First Number",strlen("Server-   Enter
the First Number : "));
read(newsockfd,&n1,sizeof(int));
printf("Client-   First Number  is : %d\n",n1);
write(newsockfd,"Server-   Enter the Second Number",strlen("Server-   Enter
the Second Number : "));
read(newsockfd,&n2,sizeof(int));
printf("Client-   Second  Number  is : %d\n",n2);
```

```
M :
switch(choice)
{
      case 1:
            ans=n1+n2;
            break;
      case 2:
            ans=n1-n2;
            break;
      case 3:
            ans=n1*n2;
            break;
      case 4:
            ans=n1/n2;
            break;
      case 5:
            goto Q;
            break;
}
write(newsockfd,&ans,sizeof(int));
write(newsockfd,"Server-   Do You Want More Arithmetic(1/0)  ?
",strlen("Server-   Do You Want More Arithmetic(1/0)  ? "));
read(newsockfd,&yes,sizeof(int));
if(yes==1)
{
printf("Client-   I want  More Arithmetic \n");
goto S;
}

Q :   write(newsockfd,"Server-   Exited\n",strlen("Server-   Exited\n"));
      printf("Client-   Exited\n");
      close(newsockfd);
      close(sockfd);

return 0;
}
```

*****OUTPUT*****

```
Aj:~$ cc server3.c -o s3
Aj:~$ ./s3 60005
Client-   The choice is : 1
Client-   First Number  is : 4
Client-   Second  Number  is : 5
Client-   I want  More Arithmetic
Client-   The choice is : 3
Client-   First Number  is : 6
Client-   Second  Number  is : 7
Client-   Exited
Aj:~$
```

(Trigonometric Calculator)

TCP Scientific Calci

//CLIENT SIDE

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <netdb.h>
#include<arpa/inet.h>
#include<unistd.h>
#include<math.h>

void error(const char *msg)
{
    perror(msg);
    exit(0);
}

int main(int argc, char *argv[])
{
    int sockfd, portno, n;
    struct sockaddr_in serv_addr;
    struct hostent *server; char fname[25];
    int choice,yes;
    float angle,ans;
    char buffer[256];
    char s_num[5];
      int num;
    if (argc < 3) {
       fprintf(stderr,"usage %s hostname port\n", argv[0]);
       exit(0);
    }
    portno = atoi(argv[2]);
    sockfd = socket(AF_INET, SOCK_STREAM, 0);
    if (sockfd < 0)
        error("ERROR opening socket");
    server = gethostbyname(argv[1]);
    if (server == NULL) {
        fprintf(stderr,"ERROR, no such host\n");
        exit(0);
    }
```

```
    bzero((char *) &serv_addr, sizeof(serv_addr));
    serv_addr.sin_family = AF_INET;
    bcopy((char *)server->h_addr,
         (char *)&serv_addr.sin_addr.s_addr,
         server->h_length);
    serv_addr.sin_port = htons(portno);
    if (connect(sockfd,(struct sockaddr *) &serv_addr,sizeof(serv_addr)) <
0)
        error("ERROR connecting");
A : bzero(buffer,256);
read(sockfd,buffer,256);
printf("%s\n",buffer);
bzero(buffer,256);
scanf("%d",&choice);
write(sockfd,&choice,sizeof(int));
if(choice==4)
goto T;
read(sockfd,buffer,256);
printf("%s\n",buffer);
bzero(buffer,256);
scanf("%f",&angle);
write(sockfd,&angle,sizeof(float));

read(sockfd,&ans,sizeof(float));
//ans=round(ans);
printf("Server-   The Answer is : %.2f\n",ans);
read(sockfd,buffer,256);
printf("%s\n",buffer);
scanf("%d",&yes);
write(sockfd,&yes,sizeof(int));
if(yes==1)
goto A;

T :   bzero(buffer,256);
      read(sockfd,buffer,256);
      printf("%s",buffer);
      close(sockfd);
return 0;
}
```

*****OUTPUT*****

```
Aj:~$ cc client4.c -o c4 -lm
Aj:~$ ./c4 192.168.0.109 65009
Server-     *******TRIGO - CALCULATOR*******
1. SINE
2. COSINE
3. TANGENT
4. EXIT

ENTER  YOUR  CHOICE
```

```
1
Server-   Enter the ANGLE
60
Server-   The Answer is : 0.87
Server-   Do You Want More Calculation(1/0)  ?
1
Server-     *******TRIGO - CALCULATOR*******
1. SINE
2. COSINE
3. TANGENT
4. EXIT

ENTER  YOUR  CHOICE
2
Server-   Enter the ANGLE
60
Server-   The Answer is : 0.50
Server-   Do You Want More Calculation(1/0)  ?
1
Server-     *******TRIGO - CALCULATOR*******
1. SINE
2. COSINE
3. TANGENT
4. EXIT

ENTER  YOUR  CHOICE
3
Server-   Enter the ANGLE
60
Server-   The Answer is : 1.73
Server-   Do You Want More Calculation(1/0)  ?
1
Server-     *******TRIGO - CALCULATOR*******
1. SINE
2. COSINE
3. TANGENT
4. EXIT

ENTER  YOUR  CHOICE
2
Server-   Enter the ANGLE
90
Server-   The Answer is : 0.00
Server-   Do You Want More Calculation(1/0)  ?
1
Server-     *******TRIGO - CALCULATOR*******
1. SINE
2. COSINE
3. TANGENT
4. EXIT
```

```
ENTER  YOUR  CHOICE
4
Server-   Exited
```

//SERVER SIDE

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include<arpa/inet.h>
#include<math.h>
#define PI  3.14159

void error(const char *msg)
{
    perror(msg);
    exit(1);
}

int main(int argc, char *argv[])
{
     int sockfd, newsockfd, portno;
     socklen_t clilen;
     char menu[100]="NO VALUE";
     char buffer[256],m[5];
     struct sockaddr_in serv_addr, cli_addr;
     int n,choice,yes;
     float angle,ans;
     if (argc < 2) {
         fprintf(stderr,"ERROR, no port provided\n");
         exit(1);
     }
     sockfd = socket(AF_INET, SOCK_STREAM, 0);
     if (sockfd < 0)
        error("ERROR opening socket");
     bzero((char *) &serv_addr, sizeof(serv_addr));
     portno = atoi(argv[1]);
     serv_addr.sin_family = AF_INET;
     serv_addr.sin_addr.s_addr = INADDR_ANY;
     serv_addr.sin_port = htons(portno);
     if (bind(sockfd, (struct sockaddr *) &serv_addr,
              sizeof(serv_addr)) < 0)
```

```
               error("ERROR on binding");
     listen(sockfd,5);
     clilen = sizeof(cli_addr);
     newsockfd = accept(sockfd,
                 (struct sockaddr *) &cli_addr,
                 &clilen);
     if (newsockfd < 0)
          error("ERROR on accept");
     bzero(buffer,256);

S : strcpy(menu,"Server-     *******TRIGO - CALCULATOR*******\n1. SINE\n2.
COSINE\n3. TANGENT\n4. EXIT\n\nENTER  YOUR  CHOICE");
write(newsockfd,menu,strlen(menu));
read(newsockfd,&choice,sizeof(int));
printf("Client-   The choice is : %d\n",choice);
if(choice==4)
goto M;
write(newsockfd,"Server-   Enter the ANGLE",strlen("Server-   Enter the
ANGLE"));
read(newsockfd,&angle,sizeof(float));
printf("Client-   Angle  is : %f\n",angle);

M :
switch(choice)
{
      case 1:
            ans=sin((angle/180)*PI);
            break;
      case 2:
            ans=cos((angle/180)*PI);
            break;
      case 3:
            ans=tan((angle/180)*PI);
            break;
      case 4:
            goto Q;
            break;
}
write(newsockfd,&ans,sizeof(float));
write(newsockfd,"Server-   Do You Want More Calculation(1/0)  ?
",strlen("Server-   Do You Want More Calculation(1/0)  ? "));
read(newsockfd,&yes,sizeof(int));
if(yes==1)
{
printf("Client-   I want  More Calculation \n");
goto S;
}

Q :   write(newsockfd,"Server-   Exited\n",strlen("Server-   Exited\n"));
      printf("Client-   Exited\n");
      close(newsockfd);
```

```
      close(sockfd);

return 0;
}
```

*****OUTPUT*****

```
Aj:~$ cc server4.c -o s4 -lm
Aj:~$ ./s4 65009
Client-   The choice is : 1
Client-   Angle  is : 60.000000
Client-   I want  More Calculation
Client-   The choice is : 2
Client-   Angle  is : 60.000000
Client-   I want  More Calculation
Client-   The choice is : 3
Client-   Angle  is : 60.000000
Client-   I want  More Calculation
Client-   The choice is : 2
Client-   Angle  is : 90.000000
Client-   I want  More Calculation
Client-   The choice is : 4
Client-   Exited
```

ASSIGNMENT:9 (Multiuser Chat)

Server:

```
import java.io.DataInputStream;
import java.io.PrintStream;
import java.io.IOException;
import java.net.Socket;
import java.net.ServerSocket;

public class Server {


  private static ServerSocket serverSocket = null;

  private static Socket clientSocket = null;


  private static final int maxClientsCount = 10;
  private static final clientThread[] threads = new
clientThread[maxClientsCount];

  public static void main(String args[]) {


    int portNumber = 6002;
    if (args.length < 1) {
      System.out.println("Usage: java MultiThreadChatServerSync
<portNumber>\n"
          + "Now using port number=" + portNumber);
    } else {
      portNumber = Integer.valueOf(args[0]).intValue();
    }

    try {
      serverSocket = new ServerSocket(portNumber);
    } catch (IOException e) {
      System.out.println(e);
    }

    while (true) {
      try {
        clientSocket = serverSocket.accept();
        int i = 0;
        for (i = 0; i < maxClientsCount; i++) {
          if (threads[i] == null) {
            (threads[i] = new clientThread(clientSocket, threads)).start();
            break;
          }
        }
```

```
        if (i == maxClientsCount) {
          PrintStream os = new PrintStream(clientSocket.getOutputStream());
          os.println("Server too busy. Try later.");
          os.close();
          clientSocket.close();
        }
      } catch (IOException e) {
        System.out.println(e);
      }
    }
  }
}


class clientThread extends Thread {

  private String clientName = null;
  private DataInputStream is = null;
  private PrintStream os = null;
  private Socket clientSocket = null;
  private final clientThread[] threads;
  private int maxClientsCount;

  public clientThread(Socket clientSocket, clientThread[] threads) {
    this.clientSocket = clientSocket;
    this.threads = threads;
    maxClientsCount = threads.length;
  }

  public void run() {
    int maxClientsCount = this.maxClientsCount;
    clientThread[] threads = this.threads;

    try {

      is = new DataInputStream(clientSocket.getInputStream());
      os = new PrintStream(clientSocket.getOutputStream());
      String name;
      while (true) {
        os.println("Enter your name.");
        name = is.readLine().trim();
        if (name.indexOf('@') == -1) {
          break;
        } else {
          os.println("The name should not contain '@' character.");
        }
      }


      os.println("Welcome " + name
          + " to our chat room.\nTo leave enter /quit in a new line.");
```

```
synchronized (this) {
  for (int i = 0; i < maxClientsCount; i++) {
    if (threads[i] != null && threads[i] == this) {
      clientName = "@" + name;
      break;
    }
  }
  for (int i = 0; i < maxClientsCount; i++) {
    if (threads[i] != null && threads[i] != this) {
      threads[i].os.println("*** A new user " + name
          + " entered the chat room !!! ***");
    }
  }
}

while (true) {
  String line = is.readLine();
  if (line.startsWith("/quit")) {
    break;
  }

  if (line.startsWith("@")) {
    String[] words = line.split("\\s", 2);
    if (words.length > 1 && words[1] != null) {
      words[1] = words[1].trim();
      if (!words[1].isEmpty()) {
        synchronized (this) {
          for (int i = 0; i < maxClientsCount; i++) {
            if (threads[i] != null && threads[i] != this
                && threads[i].clientName != null
                && threads[i].clientName.equals(words[0])) {
              threads[i].os.println("<" + name + "> " + words[1]);

              this.os.println(">" + name + "> " + words[1]);
              break;
            }
          }
        }
      }
    }
  } else {
    synchronized (this) {
      for (int i = 0; i < maxClientsCount; i++) {
        if (threads[i] != null && threads[i].clientName != null) {
          threads[i].os.println("<" + name + "> " + line);
        }
      }
    }
  }
}
synchronized (this) {
```

```
        for (int i = 0; i < maxClientsCount; i++) {
          if (threads[i] != null && threads[i] != this
              && threads[i].clientName != null) {
            threads[i].os.println("*** The user " + name
                + " is leaving the chat room !!! ***");
          }
        }
      }
      os.println("*** Bye " + name + " ***");

      synchronized (this) {
        for (int i = 0; i < maxClientsCount; i++) {
          if (threads[i] == this) {
            threads[i] = null;
          }
        }
      }

      is.close();
      os.close();
      clientSocket.close();
    } catch (IOException e) {
    }
  }
}
```

Client:

```
import java.io.DataInputStream;
import java.io.PrintStream;
import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.io.IOException;
import java.net.Socket;
import java.net.UnknownHostException;

public class TcpClient1 implements Runnable {


  private static Socket clientSocket = null;

  private static PrintStream os = null;

  private static DataInputStream is = null;

  private static BufferedReader inputLine = null;
  private static boolean closed = false;

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

```
    int portNumber = 6002;

    String host = "localhost";

    if (args.length < 2) {
      System.out
          .println("Usage: java MultiThreadChatClient <host>
<portNumber>\n"
              + "Now using host=" + host + ", portNumber=" + portNumber);
    } else {
      host = args[0];
      portNumber = Integer.valueOf(args[1]).intValue();
    }

    try {
      clientSocket = new Socket(host, portNumber);
      inputLine = new BufferedReader(new InputStreamReader(System.in));
      os = new PrintStream(clientSocket.getOutputStream());
      is = new DataInputStream(clientSocket.getInputStream());
    } catch (UnknownHostException e) {
      System.err.println("Don't know about host " + host);
    } catch (IOException e) {
      System.err.println("Couldn't get I/O for the connection to the host "
          + host);
    }

    if (clientSocket != null && os != null && is != null) {
      try {


        new Thread(new TcpClient1()).start();
        while (!closed) {
          os.println(inputLine.readLine().trim());
        }
        os.close();
        is.close();
        clientSocket.close();
      } catch (IOException e) {
        System.err.println("IOException:  " + e);
      }
    }
  }

  public void run() {

    String responseLine;
    try {
      while ((responseLine = is.readLine()) != null) {
        System.out.println(responseLine);
        if (responseLine.indexOf("*** Bye") != -1)
```

```
          break;
      }
      closed = true;
    } catch (IOException e) {
      System.err.println("IOException:  " + e);
    }
  }
}
```

Output:

Server:

```
aj@aj:~$ cd Downloads/
aj@aj:~/Downloads$ ls
android-studio      Server.java      tcpserver.java
as3dbmstheory.docx  TcpClient1.java  UDPClient.java
eclipse-installer   tcpclient.java   UDPServer.java
aj@aj:~/Downloads$ javac Server.java
Note: Server.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
aj@aj:~/Downloads$ java Server
Usage: java MultiThreadChatServerSync <portNumber>
Now using port number=6002
```

Client1:

```
aj@aj:~$ cd Downloads/
aj@aj:~/Downloads$ javac TcpClient1.java
Note: TcpClient1.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
aj@aj:~/Downloads$ java TcpClient1
Usage: java MultiThreadChatClient <host> <portNumber>
Now using host=localhost, portNumber=6002
Enter your name.
Aj
Welcome Aj to our chat room.
To leave enter /quit in a new line.
Hii,this is aj
<Aj> Hii,this is aj
*** A new user Milind entered the chat room !!! ***
<Milind> Milind Here
Welcome
<Aj> Welcome
*** A new user Mahesh entered the chat room !!! ***
<Mahesh> Hello!!!!
<Milind> hii
bye
<Aj> bye
```

Client2:

```
aj@aj:~/Downloads$ java TcpClient1
Usage: java MultiThreadChatClient <host> <portNumber>
Now using host=localhost, portNumber=6002
Enter your name.
Milind
Welcome Milind to our chat room.
To leave enter /quit in a new line.
Milind Here
<Milind> Milind Here
<Aj> Welcome
*** A new user Mahesh entered the chat room !!! ***
<Mahesh> Hello!!!!
hii
<Milind> hii
<Aj> bye
```

Client3:

```
aj@aj:~$ cd Downloads/
aj@aj:~/Downloads$ javac TcpClient1.java
Note: TcpClient1.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
aj@aj:~/Downloads$ java TcpClient1
Usage: java MultiThreadChatClient <host> <portNumber>
Now using host=localhost, portNumber=6002
Enter your name.
Mahesh
Welcome Mahesh to our chat room.
To leave enter /quit in a new line.
Hello!!!!
<Mahesh> Hello!!!!
<Milind> hii
<Aj> bye
```

ASSIGNMENT-13

Source Code:

```
import java.net.InetAddress;
import java.net.UnknownHostException;
import java.util.Scanner;

public class DnsNettwork {
	public static void main(String[] args) throws UnknownHostException {
		Scanner in =new Scanner(System.in);
		InetAddress objInet = InetAddress.getLocalHost();

		System.out.println("Hostname is : "+objInet);


		System.out.print("\nEnter site domain name : ");
		String siteName=in.next();
		System.out.println("site address is(getByName) :
"+InetAddress.getByName(siteName));


		System.out.println("\nsite address is(getAllByName) : ");
		InetAddress allAddressByName[]=
InetAddress.getAllByName(siteName);
		for(InetAddress a:allAddressByName)
		{
			System.out.println(a);
		}


		System.out.print("\nEnter IP Address : ");
		String addr=in.next();
		InetAddress ip=InetAddress.getByName(addr);


		System.out.print("Get by Address : ");

		System.out.println(ip.getHostName());
		in.close();
	}
}
```

Output:

```
Hostname is : AJAY/127.0.1.1

Enter site domain name : www.google.com
site address is(getByName) : www.google.com/74.125.204.147

site address is(getAllByName) :
```

```
www.google.com/74.125.204.147
www.google.com/74.125.204.99
www.google.com/74.125.204.103
www.google.com/74.125.204.104
www.google.com/74.125.204.105
www.google.com/74.125.204.106
www.google.com/2404:6800:4008:c04:0:0:0:6a

Enter IP Address : 8.8.8.8
Get by Address : google-public-dns-a.google.com
```

Assignment No. 02

(Message Transfer)

//Receiver

```
/**************************************************

file: demo_rx.c
purpose: simple demo that receives characters from
the serial port and print them on the screen,
exit the program by pressing Ctrl-C

compile with the command: gcc demo_rx.c rs232.c -Wall -Wextra -o2 -o
test_rx

**************************************************/

#include <stdlib.h>
#include <stdio.h>
#include <unistd.h>
#include "rs232.h"



int main()
{
  int i, n,
      cport_nr=0,        /* /dev/ttyS0 (COM1 on windows) */
      bdrate=9600;       /* 9600 baud */

  unsigned char buf[10000];

  char mode[]={'8','N','1',0};


  if(RS232_OpenComport(cport_nr, bdrate, mode))
  {
    printf("Can not open comport\n");

    return(0);

  }


 while(1)
 {
```

```
   n = RS232_PollComport(cport_nr, buf, 10000);

    if(n > 0)
    {
      buf[n] = 0;

      for(i=0; i < n; i++)
      {
        if(buf[i] < 32)
        {
          buf[i] = '.';
        }
      }

      printf("received %i bytes: %s\n", n, (char *)buf);

    }

    usleep(100000);
 }

  return(0);
}
```

//Sender

```
/**************************************************

file: demo_tx.c
purpose: simple demo that transmits characters to
the serial port and print them on the screen,
exit the program by pressing Ctrl-C

compile with the command: gcc demo_tx.c rs232.c -Wall -Wextra -o2 -o
test_tx

**************************************************/

#include <stdlib.h>
#include <stdio.h>
#include <unistd.h>
#include "rs232.h"
```

```
int main()
{
  int i=0,
      cport_nr=0,        /* /dev/ttyS0 (COM1 on windows) */
      bdrate=9600; /* 9600 baud */
char buffer[10000];


  char mode[]={'8','N','1',0};
      printf("Enter string to be sent\n");
      gets(buffer);


 // strcpy(str[0], "The quick brown fox jumped over the lazy grey dog.\n");

 // strcpy(str[1], "Happy serial programming!\n");

  if(RS232_OpenComport(cport_nr, bdrate, mode))
  {
    printf("Can not open comport\n");

    return(0);
  }

    RS232_cputs(cport_nr, buffer);

    printf("sent: %s\n", buffer);


    usleep(1000000);  /* sleep for 1 Second */



  return(0);
}
```

(File Transfer)
//Receiver

```
/**************************************************

file: demo_rx.c
purpose: simple demo that receives characters from
the serial port and print them on the screen,
exit the program by pressing Ctrl-C
```

```
compile with the command: gcc demo_rx.c rs232.c -Wall -Wextra -o2 -o
test_rx

**************************************************/

#include <stdlib.h>
#include <stdio.h>
#include <unistd.h>
#include "rs232.h"



int main()
{
  int i, n,
      cport_nr=0,        /* /dev/ttyS0 (COM1 on windows) */
      bdrate=9600;       /* 9600 baud */

  unsigned char buf[10000];

  char mode[]={'8','N','1',0};


  if(RS232_OpenComport(cport_nr, bdrate, mode))
  {
    printf("Can not open comport\n");

    return(0);

  }


 while(1)
 {
   n = RS232_PollComport(cport_nr, buf, 10000);

    if(n > 0)
    {
      buf[n] = 0;

      for(i=0; i < n; i++)
      {
        if(buf[i] < 32)
        {
```

```
          buf[i] = '.';
        }
      }

      printf("received %i bytes: %s\n", n, (char *)buf);


FILE *f;
f=fopen("input.txt","a");


  fprintf(f,buf);
fclose(f);

    }

    usleep(100000);
 }

  return(0);
}
```

//Sender

```
/**************************************************

file: demo_tx.c
purpose: simple demo that transmits characters to
the serial port and print them on the screen,
exit the program by pressing Ctrl-C

compile with the command: gcc demo_tx.c rs232.c -Wall -Wextra -o2 -o
test_tx

**************************************************/

#include <stdlib.h>
#include <stdio.h>
#include <unistd.h>
#include "rs232.h"


int main()
{
  int i=0,
```

```
      cport_nr=0,        /* /dev/ttyS0 (COM1 on windows) */
      bdrate=9600; /* 9600 baud */
char buffer[10000];


  char mode[]={'8','N','1',0},
       str[2][512];

 FILE *f;
     f=fopen("newinput.txt","r");

if (f)
{
  fseek (f, 0, SEEK_END);
  int length=ftell (f);
  fseek (f, 0, SEEK_SET);
    fread (buffer, 1, length, f);
}
  /*strcpy(str[0], "The quick brown fox jumped over the lazy grey dog.\n");

  strcpy(str[1], "Happy serial programming!\n");*/

  if(RS232_OpenComport(cport_nr, bdrate, mode))
  {
    printf("Can not open comport\n");

    return(0);
  }

    RS232_cputs(cport_nr, buffer);

    printf("sent: %s\n", buffer);

    usleep(1000000);  /* sleep for 1 Second */

  return(0);
}
```