Page No.
Date

Aim - WAP to demostrate group communication

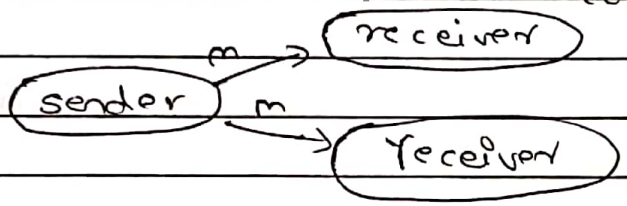
Theory -

* Group communication :-

A group is a collection of processes that act together in some system or user specified way

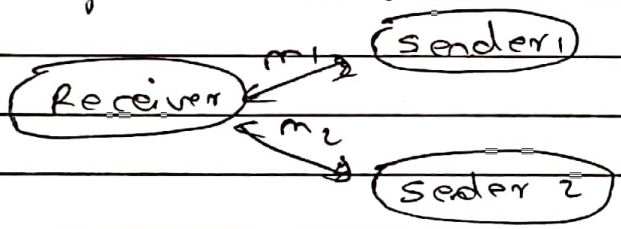
The key property that all groups have that when a message is sent to the group, itself all the members of the group receive it.

(1) Unicast :- One to one communication



Same message M sent to all nodes

(2) Many to one communication



Multiple senders send message to the same receiver

(3) Broadcast communication :-

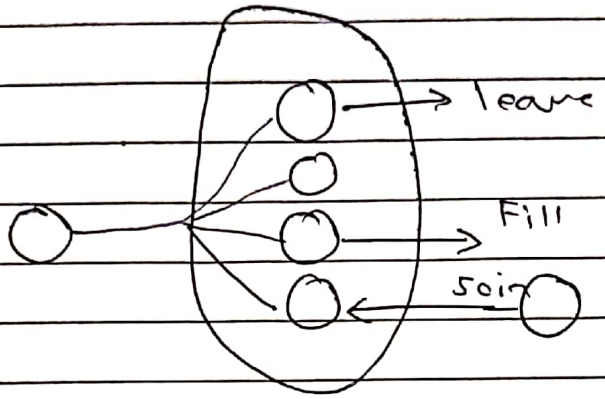
A broadcast sends a message to all the recipients

Teacher's Sign.: _____

Page No.
Date

* Group management

(1) centralized approach
(2) Distributed approach



* Description of code.

A multithreaded client/server chat application based on the console which uses Java Socket programming and multi-threading

Server

1) listens on the n/w & waits for any client to connect

2) once a client is connected can send message to the client or recive from the client

Note - Multiple client can be connected.

Teacher's Sign.: _____

Page No.
Date

Client -

1) Connects to the server on local port no.
2) Can send & receive message from the server.

* Conclusion :- We understood the concept of group communication & implemented in java successfully.

Teacher's Sign.: _____

ServerChat.java

```
import java.io.IOException;
import java.net.ServerSocket;
import java.net.Socket;
public class ServerChat{
        public static void main(String args[]){
                try(ServerSocket server =
new ServerSocket(8080)){
                        while(true){
                        Socket soc =
server.accept();
                        new
ChatAppInput(soc).start();
                        new
ChatAppOutput(soc).start();
        }
                }catch(IOException e){
       System.out.println("Issue :" + e);
                }
        }}
```

ClientChat.java

```
import java.io.IOException;
import java.net.Socket;
public class ClientChat{
        public static void main(String args[]){
                try{
                        Socket socket = new
Socket("localhost",8080);
                        new
ChatAppInput(socket).start();
                        new
ChatAppOutput(socket).start();
                }catch(IOException e){
       System.out.println("Issue :" + e);
                }catch(Exception e){

System.out.println("Issue: "+e);
                }
}
}
```

ChatAppInput.java

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.net.Socket;

class ChatAppInput extends Thread{
        private Socket socket;
        ChatAppInput(Socket socket){
                this.socket=socket;
        }
        public void run(){
                try{
       BufferedReader input = new
BufferedReader(new
InputStreamReader(socket.getInputStream(
)));
        while(true){
         String in = input.readLine();
         if(in.equals("exit")){
                break;
         }
         System.out.println("\t\t" + in);
        }
    }catch(IOException e){
        System.out.println("Issue:" +e);
    }catch(Exception e){

System.out.println("Issue :" + e);
                }
                finally{
                        try{
                                socket.close();
                        }catch(IOException
e){
            System.out.println("Issue :" +e);
                        }
                }
        }

}
```

ChatAppOutput.java

```
import java.io.IOException;
import java.io.PrintWriter;
import java.net.Socket;
import java.util.Scanner;
```

```
import java.io.BufferedReader;
import java.io.InputStreamReader;

class ChatAppOutput extends Thread{
private Socket socket;
        ChatAppOutput(Socket socket){
                this.socket=socket;
        }
        public void run(){
                        BufferedReader br =
null;
                try{
        PrintWriter output=new
PrintWriter(socket.getOutputStream(), true);
         br = new BufferedReader(new
InputStreamReader(System.in));
        //Scanner scanner = new
Scanner(System.in);

        while(true){
        //System.out.println("type
something");
        //String str = scanner.nextLine();
        String str = br.readLine();
        output.println(str);

}
                }catch(IOException e){
        System.out.println("Issue:" + e);
                }catch(Exception e){
        System.out.println("Issue:" + e);
                }
                finally{
                        try{
                                socket.close();
                        }catch(IOException
e){
            System.out.println("Issue :" +e);
                        }
                }
         }
}
```

Output

```
C:\Users\Student\Desktop\D17B_6,8>java ServerChat
                hiiiii
                hiii2
hello evry1
hello evry2
bye1
bye2
```

```
D:\Group Communication\D17B_6,8>java ClientChat
hiiiii
                hello evry1
                bye1
```

```
D:\D17B-6,8\D17B_6,8>java ClientChat
hiii2
                hello evry2
                bye2
```