Experiment no: Date:	Stop and wait protocol

Aim: To write a java program to perform stop and wait protocol.

Algorithm: step 1 : start

Step 2: Get the frame size from the user

step 3: To create the frame based on the user request

Step 4: To send frames to server from the client side.

step 5: Ack signal will be received if transmisson is successful otherwise NACK will be received.

step 6: Stop

sender:

```
import java.io.*;
import java.net.*;
class Stopandwait{
  Socket s; ObjectOutputStream out; ObjectInputStream in;
  String packet,ack,str,msg;
  int n,i=0, sequence=0;
  stopandwait(){}
  public void run(){
    try{
     BufferedReader br= new BufferedReader(new Input
                                   StreamReader(System.in));
     system.out.println("waiting for connection.....");
     s = new Socket("localhost", 2004); sequence =0;
     out = new ObjectOutputStream(s.getOutputStream());
     out.flush();
in = new ObjectInputStream(s.getInputStream());

str = (String) in.readObject();
System.out.println("receiver >"+str+"\nEnter the data tosend...");
packet = br.readLine(); n =packet.length();
 do{
  try{
   if(i<n){
      msg = String.valueOf(sequence);
      msg=msg.concat(packet.substring(i,i+1));
   }else if(i==n){
      msg ="end";out.writeObject(msg);break;
    }
   out.writeObject(msg); sequence =(sequence ==0)?1:0;
   out.flush();
   System.out.println("data sent>"+msg);
   ack =(String) in.readObject(); System.out.print("waiting for
                                                  ack...\n\n");
   if(ack.equals(String.valueOf(sequence))){
      i++;
      System.out.println("receiver >" + "packet received\n\n");
    }else{
      System.out.println("Time out resending data...\n\n");
      sequence=(sequence ==0)?1:0;
    }catch(Exception e){}
    }while(i<n+1);System.out.println("All data sent.exiting");
    }catch(Exception e){}
    finally{
       in.close();out.close();s.close();
    }
}
```

```
public static void main(String a[]){
      stopandwait s = new stopandwait();
      s.run();
   }
}
```

Receiver program:

```
import java.io.*;
import java.net.*;
class Receiver{
   ServerSocket ss; Socket conn=null; ObjectOuputStream out;
   ObjectInputStream in; String packet,ack,data="";
   int i =0, sequence =0;
   public void run(){
    try{
     BufferedReader br = new BufferedReader(new InputStreamReader
                                                   (System.in));
     ss = new ServerSocket(2004,10);
     System.out.println("waiting for connection...");
     Conn = ss.accept(); sequence =0;
     System.out.println("connection established :");
     out = new ObjectOutputStream(conn.getOutputStream());
     out.flush();
     in= new ObjectInputStream(Conn.getInputStream());
     out.writeObject("connected :");
     do{
      try{
        packet =(String)in.readObject();
        if(Integer.valueOf(packet.substring(0,1))== sequence){
          data += packet.substring(1);
          sequence =(sequence ==0)?1:0;
      }
```

sender output :

```
waiting for connection....
receiver > connected
Enter the data to send....
Hi
data sent >0 H
waiting forack...
receiver > packet received
data sent >1i
waiting for ack....
receivedr > packet received
All data sent. exiting
```

Receiver output:

```
waiting for connection...
Connection established :
receiver  >0 H
receiver  >1i
Data received : Hi
waiting for connection....
```

```
System.out.println("\n\n receiver >"+packet);
 } else {
   System.out.println("\n\n receiver >"+ packet +" duplicate data");
 } if(i<3) {
     out.writeObject(String.valueOf(sequence)); i++;
  } else {
    out.writeObject(String.valueOf((sequence+1)%2)); i=0;
    }
  } catch(Exception e){}
 }while(!packet.equals("end"));
 System.out.println("Data received ="+data);
 out.writeObject("connection ended.");
 } catch(Exception e){}
 finally{
    in.close(); out.close(); s.close();
 }}
public static void main(String args[]){
         Receiver r = new Receiver();
         while(true) r.run();
    }
}
```

Result: Thus the implementation of stop and wait protocol was implemented and executed successfully.