Experiment.No:	Sliding window Protocol
Date:	

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

Algorithm: Step 1: Start

Step 2: Get the frame based on the user request

Step 3: To send frames to server from the client side.

Step 4: If your frames reach the server ACK will be sent otherwise NACK.

Step 5: Stop.

Sender program:

```
import java.io.*;
import java.net.*;
import java.util.*;
public class slidesender{
    Socket sender; ObjectOutputStream out;
    ObjectInputStream in; String pkt; char data='a';
    int SeqNum=1, SWS=5; int LAR=0, LFS=0, NF;
    public void SendFrames(){
        if((SeqNum<=15)&&(SWS>(LFS-LAR))){
            try{
                NF=SWS-(LFS-LAR);
                for(int i=0; i<NF; i++){
                    pkt=String.valueOf(SeqNum);
                pkt=pkt.concat(" ");
                pkt=pkt.concat(String.valueOf(data));
                out.writeObject(pkt); LFS=SeqNum;
                System.out.println("sent "+SeqNum+" "+data);
```

```
    data++; if(data == 'f') data = 'a'; seqNum++;
    out.flush();
   }
  } catch(Exception e) {}
 }
}
public void run() throws IOException {
     sender = new Socket("localhost", 1500);
     out = new ObjectOutputStream(sender.getOutputStream());
     in = new ObjectInputStream(sender.getInputStream());
      while(LAR) {
          try {
           sendFrames();
           String ACK = (String) in.readObject();
           LAR = Integer.parseInt(ACK);
           System.out.println("ack received:" + LAR);
         } catch(Exception e) {}
       }
      in.close(); out.close(); sender.close();
      System.out.println("Connection Terminated.");
    }
  public static void main(String[] a) throws IOException {
        slidesender s = new slidesender(); s.run();
     }
  }
```

Receiver program:

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

public class slidereceiver {
  ServerSocket receiver; Socket conc = null;
```

```
ObjectOutputStream out; ObjectInputStream in;
String ack, pkt, data = " "; int delay;
int seqNum = 0, RWS = 5, LFR = 0; int LAF = LFR + RWS;
Random rand = new Random();

public void run() throws Exception{
    receiver = new ServerSocket(1500, 10);
    conc = receiver.accept();
    if (conc != null) System.out.println("connection established:");
    out = new ObjectOutputStream(conc.getOutputStream());
    in = new ObjectInputStream(conc.getInputStream());

    while(LFR < 15) {
      try{
      pkt = (String) in.readObject();
      String[] str = pkt.split("19"); ack = str[0]; data = str[1];
      LFR = Integer.parseInt(ack);
    if((seqNum <= LFR) || (seqNum > LAF)) {
      System.out.println("In Msg received : " + data);
      delay = rand.nextInt(5);
      if(delay < 3 || LFR == 15) {
        out.writeObject(ack); out.flush();
        System.out.println("sending ack " + ack);
        seqNum++;
      } else System.out.println("Not sending Ack");
    } else {
      out.writeObject(LFR); out.flush();
      System.out.println("resending ack " + LFR);
    } catch(Exception e) {}
```

```
in.close();
out.close();
receiver.close();
System.out.println("Inconnection terminated.");
}
public static void main(String[] args) throws Exception {
    SlideReceiver R = new SlideReceiver();
    R.run();
}
}
```

Result: Thus the Implementation of sliding window protocol was Implemented and executed successfully.