

A thick dark blue vertical bar runs down the left side of the page. A blue arrow points to the right from this bar, containing the date.

2/19/2022

# DCN

## Assignment -2

Several thin, curved lines in shades of blue and grey originate from the bottom left corner and sweep upwards and to the right.

3152/B Hardik Togadiya

## 1. Write a java program to implement Single Bit - Even Parity using TCP.

Client :

```
import java.io.*;
import java.net.*;
import java.util.Scanner;
class p1_client
{
    public static void main(String args[]) throws Exception
    {
        try
        {
            Socket socket=new Socket("localhost",6363);
            DataOutputStream outstream=new
DataOutputStream(socket.getOutputStream());
            String msg="",op="",finalmsg="";
            int counter=0,i,j;
            Scanner sc=new Scanner(System.in);
            System.out.println("Enter message:");
            msg=sc.nextLine();

            for(i=0;i<msg.length();i++)
            {
                op=Integer.toBinaryString(msg.charAt(i));
                counter=0;
                while(op.length()<7)
                {
                    op='0'+op;
                    for(j=0;j<op.length();j++)
                    {
                        if(op.charAt(j)=='1')
                            counter++;
                    }
                    if(counter%2==0)
                        op+='0';
                    else
                        op+='1';
                    finalmsg += op;
                    System.out.println("Character : "+op);
                }
                System.out.println("After adding parity bit:"+finalmsg);

                outstream.writeUTF(finalmsg);
                outstream.flush();
                outstream.close();
                socket.close();
            }
        }
        catch(Exception e)
        {
            System.out.println(e);
        }
    }
}
```

```

    }
}

```

## Server:

```

import java.io.*;
import java.net.*;
class p1_server
{
    public static void main(String args[])throws Exception
    {
        try
        {
            ServerSocket serversocket = new ServerSocket(6363);
            Socket socket=serversocket.accept();
            DataInputStream instream=new
DataInputStream(socket.getInputStream());
            DataOutputStream outstream=new
DataOutputStream(socket.getOutputStream());

            String finalmsg=instream.readUTF();
            int i,j,k,counter=0;
            String f="",op="";

            for(i=0;i<finalmsg.length();i+=8)
            {
                String decode=finalmsg.substring(i,i+8);
                System.out.println("Message Checking:"+decode);
                for(j=0;j<decode.length();j++)
                {
                    if(decode.charAt(j)=='1')
                        counter++;
                }
                if(counter%2==0)
                    System.out.println("Correct message passes of character :
"+((i/8)+1));
                else
                {
                    System.out.println("Please send the message again..");
                    System.exit(0);
                }
            }
            System.out.println("Message received successfully...");
            for(k=0;k<finalmsg.length();k+=8)
            {
                op=finalmsg.substring(k,k+7);
                System.out.println("After removing parity bit:"+op);
                f+=(char)Byte.parseByte(op,2);
            }
            System.out.println("Final Message:"+f);

            instream.close();
            socket.close();
            serversocket.close();

```

```
    }  
    catch(Exception e)  
    {  
        System.out.println(e);  
    }  
}  
}
```

## Client:

```
PS D:\Project\DCN\Assignment - 2> java .\p1_client-1.java  
Enter message:  
hello  
Character : 11010001  
Character : 11001010  
Character : 11011000  
Character : 11011000  
Character : 11011110  
After adding parity bit:1101000111001010110110001101100011011110  
PS D:\Project\DCN\Assignment - 2> java .\p1_client-1.java  
Enter message:  
hello  
Character : 11010001  
Character : 11001010  
Character : 11011000  
Character : 11011000  
Character : 11011110  
After adding parity bit:1101000111001010110110001101100011011110  
PS D:\Project\DCN\Assignment - 2> █
```

Server:

```
PS D:\Project\DCN\Assignment - 2> java .\p1_server-1.java
Message Checking:11010001
Correct message passes of character : 1
Message Checking:11001010
Correct message passes of character : 2
Message Checking:11011000
Correct message passes of character : 3
Message Checking:11011000
Correct message passes of character : 4
Message Checking:11011110
Correct message passes of character : 5
Message received successfully...
After removing parity bit:1101000
After removing parity bit:1100101
After removing parity bit:1101100
After removing parity bit:1101100
After removing parity bit:1101111
Final Message:hello
PS D:\Project\DCN\Assignment - 2> 
```

2. Write a java program to implement Single Bit - Odd Parity using UDP.

Sender:

```
import java.util.Scanner;
import java.io.*;
import java.net.*;
class p2_sender
{
    public static void main(String args[]) throws Exception
    {
        try
        {
```

```

DatagramSocket ds=new DatagramSocket();
InetAddress ip = InetAddress.getByName("localhost");

int i,j,ans,counter=0;
String msg="",op="",finalmsg="";
Scanner sc=new Scanner(System.in);
System.out.println("Enter message :");
msg=sc.nextLine();

for(i=0;i<msg.length();i++)
{
    op=Integer.toBinaryString(msg.charAt(i));
    counter=0;
    while(op.length()<7)
        op='0'+op;
    for(j=0;j<op.length();j++)
    {
        if(op.charAt(j)=='1')
            counter++;
    }
    if(counter%2!=0)
        op+='0';
    else
        op+='1';
    finalmsg += op;
    System.out.println("Character : "+op);

}
System.out.println("After adding parity bit:"+finalmsg);
DatagramPacket dp=new
DatagramPacket(finalmsg.getBytes(),finalmsg.length(),ip,6565);

ds.send(dp);

ds.close();

}
catch(Exception e)
{
    System.out.println(e);
}
}
}

```

## Reciver:

```

import java.io.*;
import java.net.*;
class p2_receiver
{
    public static void main(String args[]) throws Exception
    {
        try
        {

```

```

DatagramSocket ds=new DatagramSocket(6565);

byte[] buf = new byte[1024]; //client data mokle e aama store thasse
DatagramPacket dp=new DatagramPacket(buf,1024);//Or we can write
buf.length instead of 1024
ds.receive(dp);

int i,j,counter=0;
String msg=new String(dp.getData(),0,dp.getLength());
for(i=0;i<msg.length();i+=8)
{
    String decode=msg.substring(i,i+8);
    System.out.println("Message Checking:"+decode);
    for(j=0;j<decode.length();j++)
    {
        if(decode.charAt(j)=='1')
            counter++;
    }
    if(counter%2!=0)
        System.out.println("Correct message passes of character :
"+((i/8)+1));
    else
        System.exit(0);
    counter=0;
}

int k;
String f="",op="";
for(k=0;k<msg.length();k+=8)
{
    op=msg.substring(k,k+7);
    System.out.println("After removing parity bit:"+op);
    f+=(char)Byte.parseByte(op,2);
}

System.out.println("Client Message : " + f);
ds.close();
}
catch(Exception e)
{
    System.out.println(e);
}
}
}

```

Sender:

```

PS D:\Project\DCN\Assignment - 2> java .\p2_sender-1.java
Enter message :
hello
Character : 11010000
Character : 11001011
Character : 11011001
Character : 11011001
Character : 11011111
After adding parity bit:1101000011001011110110011101100111011111
PS D:\Project\DCN\Assignment - 2>

```

Receiver:

```

PS D:\Project\DCN\Assignment - 2> java .\p2_receiver-1.java
Message Checking:11010000
Correct message passes of character : 1
Message Checking:11001011
Correct message passes of character : 2
Message Checking:11011001
Correct message passes of character : 3
Message Checking:11011001
Correct message passes of character : 4
Message Checking:11011111
Correct message passes of character : 5
After removing parity bit:1101000
After removing parity bit:1100101
After removing parity bit:1101100
After removing parity bit:1101100
After removing parity bit:1101111
Client Message : hello
PS D:\Project\DCN\Assignment - 2>

```

3. Write a java program to implement Block Parity using TCP.



## Server:

```
import java.io.*;
import java.net.*;
import java.util.*;
class ServerBlockParity{
    public static void main(String args[])throws Exception
    {
        try
        {
            Scanner sc=new Scanner(System.in);
            ServerSocket serversocket = new ServerSocket(6001);
            System.out.println("Server Is Listening On Localhost : 6001");

            Socket socket = serversocket.accept();
            DataInputStream istream = new
DataInputStream(socket.getInputStream());
            DataOutputStream ostream = new
DataOutputStream(socket.getOutputStream());

            String rmsg = new String();
            String omsg = new String();
            rmsg=istream.readUTF();
            char cpmg[][] = new char[rmsg.length()/8][8];
            int rowerror =0 , colerror = 0;

            System.out.println("Received Binary String      : "+rmsg);

            for(int i=0;i<rmsg.length();i+=8)
            {
                String rtemp = rmsg.substring(i,i+8);
                char p = evenParity(rtemp);

                if(p == '0')
                {
                    System.out.println("Row : " + i/8 + " No Error");
                }
                else
                {
                    rowerror++;
                    System.out.println("Row : " + i/8 + " Error Detected");
                }

                cpmg[i/8] = rtemp.toCharArray();
                rtemp = rtemp.substring(0,rtemp.length()-1);
                omsg = omsg + (char)Byte.parseByte(rtemp,2);
            }
            for(int j=0;j<8;j++)
            {
                String ctemp = new String();
                for(int i=0;i<rmsg.length()/8;i++)
```

```

        {
            ctemp = ctemp + cpmg[i][j];
        }
        char p = evenParity(ctemp);

        if(p == '0')
        {
            System.out.println("Column : " + j + " No Error");
        }
        else
        {
            colerror++;
            System.out.println("Column : " + j + " Error Detected");
        }
    }
    if(rowerror == 0 && colerror == 0)
    {
        omsg = omsg.substring(0,omsg.length()-1);
        System.out.println("\nOriginal Message : "+omsg);
    }

    istream.close();
    socket.close();
    serversocket.close();

}
catch(Exception e)
{
    System.out.println(e);
}
}
public static char evenParity(String msg)
{
    int c=0;

    for(int j=0;j<msg.length();j++)
    {
        if(msg.charAt(j)=='1')
        {
            c++;
        }
    }
    //System.out.println("Counter : "+c);
    if(c%2==0)
    {
        return '0';
    }
    else
    {
        return '1';
    }
}

```

```

    }
}

```

## Client:

```

import java.io.*;
import java.net.*;
import java.util.*;
class ClientBlockParity{
    public static void main(String args[])throws Exception
    {
        try
        {
            Scanner sc = new Scanner(System.in);

            Socket socket = new Socket("localhost",6001);
            DataInputStream istream = new
DataInputStream(socket.getInputStream());
            DataOutputStream ostream = new
DataOutputStream(socket.getOutputStream());

            System.out.print("Enter Any String : ");
            String msg=sc.nextLine();
            char gpmsg[][] = new char[msg.length()+1][8];
            String sendmsg=new String();

            for(int i=0;i<msg.length();i++)
            {
                int ch=msg.charAt(i);                                // Store Each Char ASCII
Value
                String temp=Integer.toBinaryString(ch); // Convert ASCII Value to
Binary Byte
                while(temp.length()<7)
                {
                    temp = '0'+ temp;
                }
                temp = temp + evenParity(temp);    // Send String and Check Even
Bit Parity or Odd Bit Parity

                sendmsg = sendmsg + temp;
                for(int j=0;j<temp.length();j++)
                {
                    gpmsg[i][j] = temp.charAt(j);
                }

                System.out.println("Char : "+msg.charAt(i) + " ASCII :- "+ch+"
Binary String With Parity Bit - "+temp);
            }

            //System.out.println();
            for(int i = 0;i<8;i++)

```

```

        {
            String temp = new String();
            for(int j=0;j<msg.length();j++)
            {
                temp = temp + gpmsg[j][i];
            }
            gpmsg[msg.length()][i]= evenParity(temp);

            sendmsg = sendmsg + gpmsg[msg.length()][i];

            System.out.println("Column : "+i+" Parity Bit : 
"+gpmsg[msg.length()][i]);
        }
        System.out.println("Send Binary String      : "+sendmsg);

        ostream.writeUTF(sendmsg);
        ostream.flush();

        ostream.close();
        socket.close();
    }
    catch(Exception e)
    {
        System.out.println(e);
    }
}

public static char evenParity(String msg)
{
    int c=0;

    for(int j=0;j<msg.length();j++)
    {
        if(msg.charAt(j)=='1')
            c++;
    }
    if(c%2==0)
    {
        return '0';
    }
    else
    {
        return '1';
    }
}
}

```

## Server:

```
PS D:\Project\DCN\Assignment - 2> java .\ServerBlockParity.java
Server Is Listening On Localhost : 6001
Received Binary String      : 100100001100101011011000110110001101111010000100
Row : 0 No Error
Row : 1 No Error
Row : 2 No Error
Row : 3 No Error
Row : 4 No Error
Row : 5 No Error
Column : 0 No Error
Column : 1 No Error
Column : 2 No Error
Column : 3 No Error
Column : 4 No Error
Column : 5 No Error
Column : 6 No Error
Column : 7 No Error

Original Message : Hello
PS D:\Project\DCN\Assignment - 2> █
```

## Client:

```
PS D:\Project\DCN\Assignment - 2> java .\ClientBlockParity.java
Enter Any String : Hello
Char : H ASCII :- 72 Binary String With Parity Bit - 10010000
Char : e ASCII :- 101 Binary String With Parity Bit - 11001010
Char : l ASCII :- 108 Binary String With Parity Bit - 11011000
Char : l ASCII :- 108 Binary String With Parity Bit - 11011000
Char : o ASCII :- 111 Binary String With Parity Bit - 11011110
Column : 0 Parity Bit : 1
Column : 1 Parity Bit : 0
Column : 2 Parity Bit : 0
Column : 3 Parity Bit : 0
Column : 4 Parity Bit : 0
Column : 5 Parity Bit : 1
Column : 6 Parity Bit : 0
Column : 7 Parity Bit : 0
Send Binary String          : 100100001100101011011000110110001101111010000100
PS D:\Project\DCN\Assignment - 2> █
```