



Department of Computer Science and engineering(Iot, Cyber security with block chain technology)

Class: S.Y. B.Tech.

Semester: IV

Course Code: DJS22ICL402

Course Name: Computer Networks Lab

Name: Divy Mav

SAP ID:60019220133

Experiment No: 6

Aim: Write a program to implement Error Detection and correction mechanism:

Hamming code

CRC

Code:

DATAGRAM

```
import java.net.*; class WriteServer { public
static int serverPort = 666; public static int
clientPort = 999; public static int buffer_size =
1024; public static DatagramSocket ds; public
static byte buffer[] = new byte[buffer_size];
public static void TheServer() throws Exception {
int pos=0; while (true) { int c = System.in.read();
switch (c) { case -1:
System.out.println("Server Quits.");
return; case '\r': break; case '\n':
ds.send(new
DatagramPacket(buffer,pos,InetAddress.getLocalHost(),clientPort)
);
pos=0; break; default:
buffer[pos++] = (byte)
c;
}
}
}
public static void TheClient() throws Exception {
Course Code: DJS22ICL402 Course Name: Computer Networks Lab
while(true) {
```



Department of Computer Science and engineering(Iot, Cyber security with block chain technology)

Class: S.Y. B.Tech.

Semester: IV

```
DatagramPacket p = new DatagramPacket(buffer, buffer.length);
ds.receive(p);
System.out.println(new String(p.getData(), 0, p.getLength()));
}
}
public static void main(String args[]) throws Exception {
if(args.length == 1) {
ds = new DatagramSocket(serverPort);
TheServer();
} else {
ds = new DatagramSocket(clientPort);
TheClient();
}
}
}
```

TCP Server :- import
java.io.*; import
java.net.*;

```
public class TCPServer { public static
    void main(String[] args) { try {
        ServerSocket serverSocket = new ServerSocket(12345); //
Create server socket
        System.out.println("TCP Server running...");
while (true) {
            Socket clientSocket = serverSocket.accept(); // Wait for
client connection
            System.out.println("Client connected: " +
clientSocket.getInetAddress().getHostAddress());
```



Department of Computer Science and engineering(Iot, Cyber security with block chain technology)

Class: S.Y. B.Tech.

Semester: IV

Course Code: DJS22ICL402

Course Name: Computer Networks Lab

```
BufferedReader in = new BufferedReader(new  
InputStreamReader(clientSocket.getInputStream())); PrintWriter  
out = new  
PrintWriter(clientSocket.getOutputStream(), true);
```

```
String message = in.readLine(); // Read message from client  
System.out.println("Received from client: " + message);
```

```
out.println("Message received: " + message); // Send  
response to client
```

```
clientSocket.close(); // Close connection with client  
}  
} catch (IOException e) {  
e.printStackTrace();  
}  
}  
}
```

TCP Client :-

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

```
public class TCPClient { public static  
void main(String[] args) { try {  
Socket socket = new Socket("localhost", 12345); //  
Connect to server  
System.out.println("Connected to TCP Server...");
```



Department of Computer Science and engineering(Iot, Cyber security with block chain technology)

Class: S.Y. B.Tech.

Semester: IV

```
BufferedReader userInput = new BufferedReader(new
InputStreamReader(System.in));
BufferedReader in = new
BufferedReader(newInputStreamReader(socket.getInputStream())
)
;

PrintWriter out = new
PrintWriter(socket.getOutputStream(), true);
Course Code: DJS22ICL402 Course Name: Computer Networks Lab
System.out.print("Enter message: ");
String message = userInput.readLine(); // Read user input

out.println(message); // Send message to server

String response = in.readLine(); // Receive response from server
System.out.println("Server response: " + response);

socket.close(); // Close connection with server
} catch (IOException e) {
e.printStackTrace();
}
}
}
```

OUTPUT:-

SERVER



Department of Computer Science and engineering(Iot, Cyber security with block chain technology)

Class: S.Y. B.Tech.

Semester: IV

Course Code: DJS22ICL402

Course Name: Computer Networks Lab

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19045.4046]
(c) Microsoft Corporation. All rights reserved.

C:\Users\djsce.student\Desktop\SOCKET>java TCPClient.java
Connected to TCP Server...
Enter message: Hi! This is from CSE(ICB) YAY!!
Server response: Message received: Hi! This is from CSE(ICB) YAY!!

C:\Users\djsce.student\Desktop\SOCKET>
```

CLIENT

```
C:\Windows\System32\cmd.exe - java TCPServer.java
C:\Users\djsce.student\Desktop\SOCKET>java TCPServer.java
TCP Server running...
Client connected: 127.0.0.1
Received from client: Hi! This is from CSE(ICB) YAY!!
```

UDP CLIENT

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

public class UDPClient { public static
    void main(String[] args) { try {
        DatagramSocket socket = new DatagramSocket(); // Create
UDP socket
        InetAddress serverAddress =
InetAddress.getByName("localhost");
```



Department of Computer Science and engineering(Iot, Cyber security with block chain technology)

Class: S.Y. B.Tech.

Semester: IV

```
BufferedReader userInput = new BufferedReader(new
InputStreamReader(System.in));

System.out.print("Enter message: ");
String message = userInput.readLine(); // Read user input

byte[] buffer = message.getBytes();
DatagramPacket packet = new DatagramPacket(buffer,
buffer.length, serverAddress, 9876); socket.send(packet); //
Send packet to server byte[] responseBuffer = new byte[1024];
```



Department of Computer Science and engineering(Iot, Cyber security with block chain technology)

Class: S.Y. B.Tech.

Semester: IV

Course Code: DJS22ICL402

Course Name: Computer Networks Lab

```
DatagramPacket responsePacket = new  
DatagramPacket(responseBuffer, responseBuffer.length);  
socket.receive(responsePacket); // Receive response from  
server
```

```
String response = new String(responsePacket.getData(), 0,  
responsePacket.getLength());  
System.out.println("Server response: " + response);
```

```
socket.close(); // Close socket  
} catch (IOException e) {  
    e.printStackTrace();  
}  
}  
}
```

UDP SERVER

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

```
public class UDPServer { public static  
    void main(String[] args) { try {  
        DatagramSocket socket = new DatagramSocket(9876); //  
Create UDP socket  
        System.out.println("UDP Server running...");  
  
        byte[] buffer = new byte[1024];  
        DatagramPacket packet = new DatagramPacket(buffer,  
buffer.length);
```



```
while (true) { socket.receive(packet); // Receive packet from
    client
        String message = new String(packet.getData(), 0,
packet.getLength());
        System.out.println("Received from client: " + message);
        InetAddress clientAddress = packet.getAddress(); int
        clientPort = packet.getPort();

        byte[] responseBuffer = ("Message received: " +
message).getBytes();
        DatagramPacket responsePacket = new
DatagramPacket(responseBuffer, responseBuffer.length,
clientAddress, clientPort);

        socket.send(responsePacket); // Send response to client
    }
} catch (IOException e) {
    e.printStackTrace();
}
}
```

OUTPUT:-

SERVER



Department of Computer Science and engineering(Iot, Cyber security with block chain technology)

Class: S.Y. B.Tech.

Semester: IV

Course Code: DJS22ICL402

Course Name: Computer Networks Lab

```
C:\Windows\System32\cmd.exe - java UDPServer.java
Microsoft Windows [Version 10.0.19045.4046]
(c) Microsoft Corporation. All rights reserved.

C:\Users\djsce.student\Desktop\SOCKET\UDP>javac UDPCClient.java

C:\Users\djsce.student\Desktop\SOCKET\UDP>javac UDPServer.java

C:\Users\djsce.student\Desktop\SOCKET\UDP>java UDPServer.java
UDP Server running...
Received from client: Message is sent from UDP protocol
```

Department of Computer Science and engineering(Iot, Cyber security with block chain technology)

Class: S.Y. B.Tech.

Semester: IV

Course Code: DJS22ICL402

Course Name: Computer Networks Lab

CLIENT

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19045.4046]
(c) Microsoft Corporation. All rights reserved.

C:\Users\djsce.student\Desktop\SOCKET\UDP>java UDPCClient.java
Enter message: Message is sent from UDP protocol
Server response: Message received: Message is sent from UDP protocol

C:\Users\djsce.student\Desktop\SOCKET\UDP>
```

Conclusion: To conclude that I have successfully implement and successfully performed Socket programming using Java Programming. I have successfully implemented both types of socket programming. I have successfully implemented UDP , TCP and Datagram for both server side and client side. I have learned about UDP,TCP and Datagram successfully.