Practical no:-17

Program Code:

1)

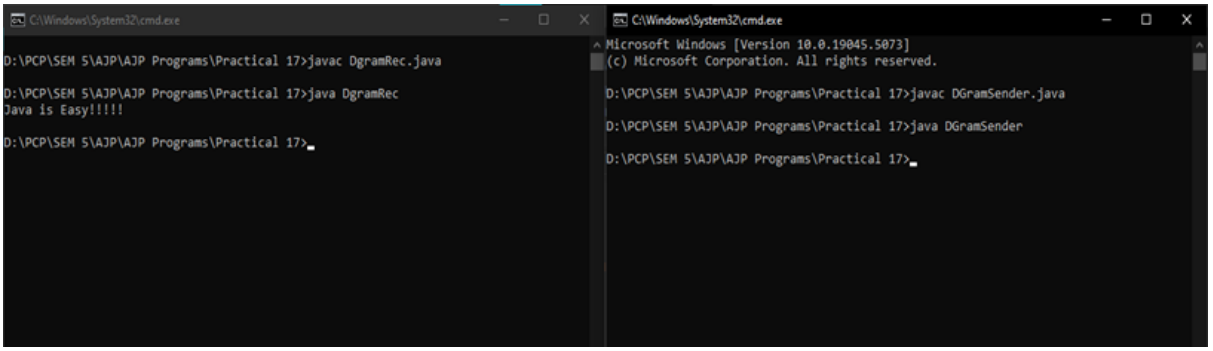
1. DgramRec.java

```
import java.net.*;
public class DgramRec {
        public static void main(String[] args) throws Exception {
        DatagramSocket ds = new DatagramSocket(3000);
        byte[] buf = new byte[1024];
        DatagramPacket dp = new DatagramPacket(buf, 1024);
        ds.receive(dp);
        String str = new String(dp.getData(), 0, dp.getLength());
        System.out.println(str);
        ds.close();
        }
}
```

2. DGramSender.java

```
import java.net.*;
public class DGramSender {
        public static void main(String[] args) throws Exception {
        DatagramSocket ds = new DatagramSocket();
        String str = "Java is Easy!!!!!";
        InetAddress ip = InetAddress.getByName("127.0.0.1");
        DatagramPacket dp = new DatagramPacket(str.getBytes(), str.length(),
```

```
                ip, 3000);
        ds.send(dp);
        ds.close();
        }
}
```



Exercise :

1)

1. ChatClient.java

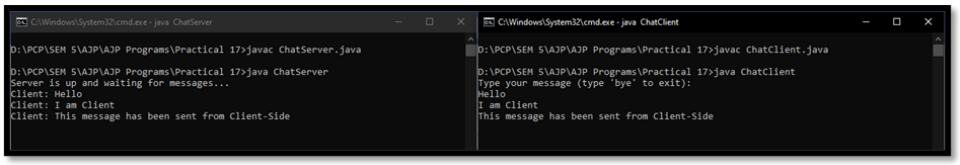
```
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;
import java.util.Scanner;
public class ChatClient {
        public static void main(String[] args) {
        try {
        DatagramSocket clientSocket = new DatagramSocket();
```

```
        InetAddress IPAddress = InetAddress.getByName("localhost");
        byte[] sendBuffer = new byte[1024];
        Scanner scanner = new Scanner(System.in);
        System.out.println("Type your message (type 'bye' to exit):");
        while (true) {
                String message = scanner.nextLine();
                sendBuffer = message.getBytes();
                DatagramPacket sendPacket = new DatagramPacket(sendBuffer,
sendBuffer.length, IPAddress, 9876);
                clientSocket.send(sendPacket);
                if (message.equalsIgnoreCase("bye")) {
              System.out.println("Chat ended.");
                break;
                }       }
        clientSocket.close();
        } catch (Exception e) {
        e.printStackTrace();
        }
        }}
```

2. ChatServer.java

```
import java.net.DatagramPacket;
import java.net.DatagramSocket;


public class ChatServer {
        public static void main(String[] args) {
```

```
        try {
        DatagramSocket serverSocket = new DatagramSocket(9876);
        byte[] receiveBuffer = new byte[1024];
        System.out.println("Server is up and waiting for messages...");
        while (true) {
                DatagramPacket receivePacket = new DatagramPacket(receiveBuffer,
receiveBuffer.length);
            serverSocket.receive(receivePacket);
                String message = new String(receivePacket.getData(), 0,
receivePacket.getLength());
            System.out.println("Client: " + message);
                if (message.equalsIgnoreCase("bye")) {
               System.out.println("Chat ended.");
                break;
                }
                receiveBuffer = new byte[1024];
        }
        serverSocket.close();
        } catch (Exception e) {
        e.printStackTrace();
        }
        }
}
```

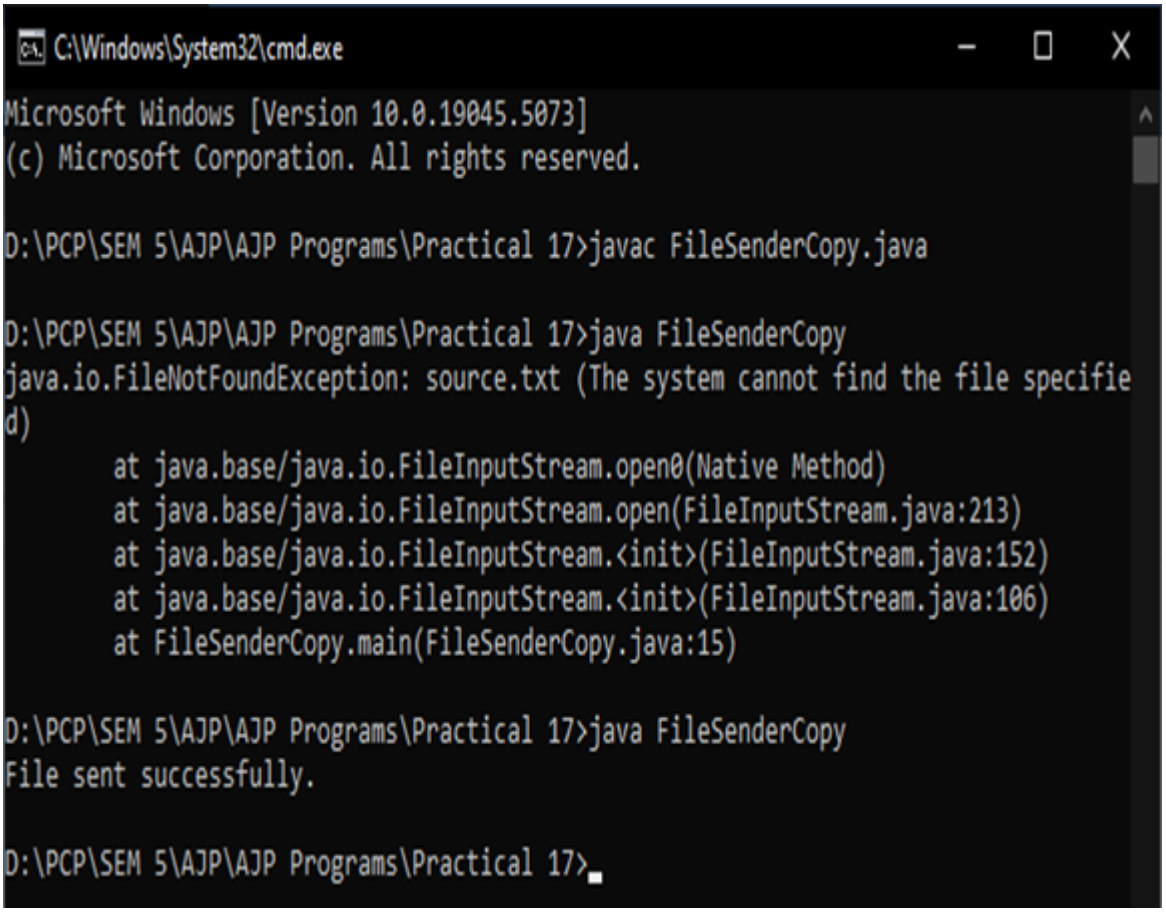


2)

1. FileSenderCopy.java

```
import java.io.FileInputStream;

import java.io.IOException;

import java.net.DatagramPacket;

import java.net.DatagramSocket;

import java.net.InetAddress;


public class FileSenderCopy {

	public static void main(String[] args) {

	final String HOST = "localhost";

	final int PORT = 12345;

	byte[] buffer = new byte[4096];


	try (DatagramSocket socket = new DatagramSocket()) {

	InetAddress address = InetAddress.getByName(HOST);

	FileInputStream fis = new FileInputStream("source.txt");

	int bytesRead;

	while ((bytesRead = fis.read(buffer)) != -1) {

		DatagramPacket packet = new DatagramPacket(buffer, bytesRead, address,
PORT);

		socket.send(packet);

	}

	fis.close();

	System.out.println("File sent successfully.");
```

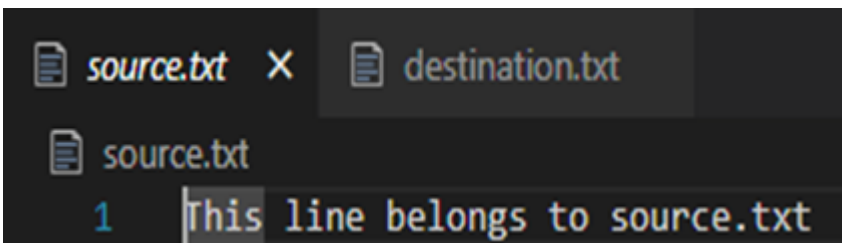
```
        } catch (IOException ex) {
        ex.printStackTrace();
        }
        }
}
```

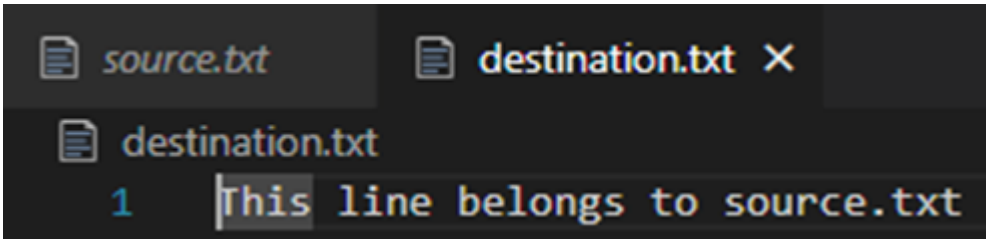


2. FileRecieverCopy.java

```
import java.io.FileOutputStream;
import java.io.IOException;
import java.net.DatagramPacket;
import java.net.DatagramSocket;
public class FileRecieverCopy {
```

```
        public static void main(String[] args) {
        final int PORT = 12345;
        byte[] buffer = new byte[4096];
        try (DatagramSocket socket = new DatagramSocket(PORT)) {
        System.out.println("Server is listening on port " + PORT);
        FileOutputStream fos = new FileOutputStream("destination.txt");
        while (true) {
                DatagramPacket packet = new DatagramPacket(buffer, buffer.length);
                socket.receive(packet);
                if (packet.getLength() == 0) {
                break;
                }
                fos.write(packet.getData(), 0, packet.getLength());
        }
        fos.close();
        System.out.println("File received successfully.");
        } catch (IOException ex) {
        ex.printStackTrace();
        }
        }
}
```



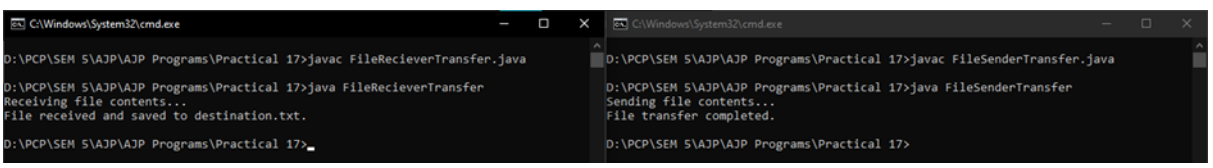


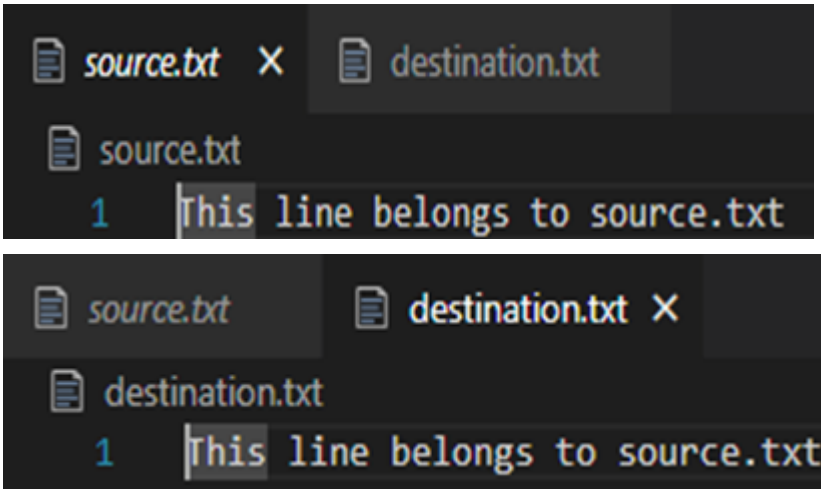
3)

1. FileSenderTransfer.java

```
import java.io.File;

import java.io.FileInputStream;

import java.net.DatagramPacket;

import java.net.DatagramSocket;

import java.net.InetAddress;


public class FileSenderTransfer {


        public static void main(String[] args) {

        try {

        File file = new File("source.txt");

        FileInputStream fis = new FileInputStream(file);


        DatagramSocket socket = new DatagramSocket();


        InetAddress receiverAddress = InetAddress.getByName("localhost");

        byte[] sendBuffer = new byte[1024];

        int bytesRead;
```

```
        System.out.println("Sending file contents...");


        while ((bytesRead = fis.read(sendBuffer)) != -1) {
                DatagramPacket sendPacket = new DatagramPacket(sendBuffer, bytesRead,
receiverAddress, 9876);
                socket.send(sendPacket);
        }


        String endSignal = "END";
        DatagramPacket endPacket = new DatagramPacket(endSignal.getBytes(),
endSignal.length(), receiverAddress, 9876);
        socket.send(endPacket);
        fis.close();
        socket.close();
        System.out.println("File transfer completed.");
        } catch (Exception e) {
        e.printStackTrace();
        }
        }
}
```





2. FileRecieverTransfer.java

```
import java.io.FileOutputStream;

import java.net.DatagramPacket;

import java.net.DatagramSocket;

public class FileRecieverTransfer {

	public static void main(String[] args) {

	try {

	DatagramSocket socket = new DatagramSocket(9876);

	byte[] receiveBuffer = new byte[1024];

	FileOutputStream fos = new FileOutputStream("destination.txt");

	System.out.println("Receiving file contents...");

	while (true) {

		DatagramPacket receivePacket = new DatagramPacket(receiveBuffer,
receiveBuffer.length);

		socket.receive(receivePacket);

		String receivedData = new String(receivePacket.getData(), 0,
receivePacket.getLength());

		if (receivedData.equals("END")) {

		break;
```

```
                }
            fos.write(receivePacket.getData(), 0, receivePacket.getLength());
        }
        fos.close();
        socket.close();
        System.out.println("File received and saved to destination.txt.");
        } catch (Exception e) {
        e.printStackTrace();
        }
        }
}
```