

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);

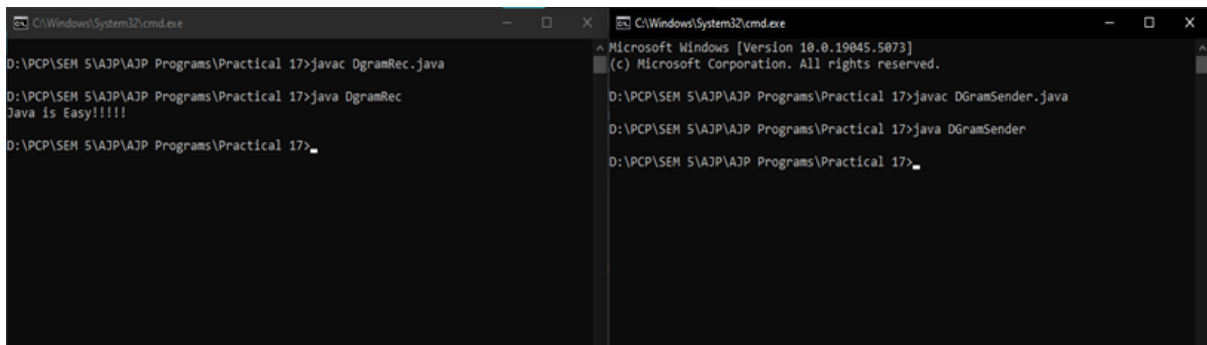
    ds.send(dp);

    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();

}

}

```

The screenshot shows two side-by-side command prompt windows. The left window, titled 'C:\Windows\System32\cmd.exe - java ChatServer', shows the execution of 'java ChatServer.java'. The output is: 'D:\PCP\SEM 5\AJPAJP Programs\Practical 17>java ChatServer.java', 'D:\PCP\SEM 5\AJPAJP Programs\Practical 17>java ChatServer', 'Server is up and waiting for messages...', 'Client: Hello', 'Client: I am Client', and 'Client: This message has been sent from Client-Side'. The right window, titled 'C:\Windows\System32\cmd.exe - java ChatClient', shows the execution of 'java ChatClient.java'. The output is: 'D:\PCP\SEM 5\AJPAJP Programs\Practical 17>java ChatClient', 'D:\PCP\SEM 5\AJPAJP Programs\Practical 17>java ChatClient', 'Type your message (type \'bye\' to exit):', 'Hello', 'I am Client', and 'This message has been sent from Client-Side'.

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);
```

```
            }
```

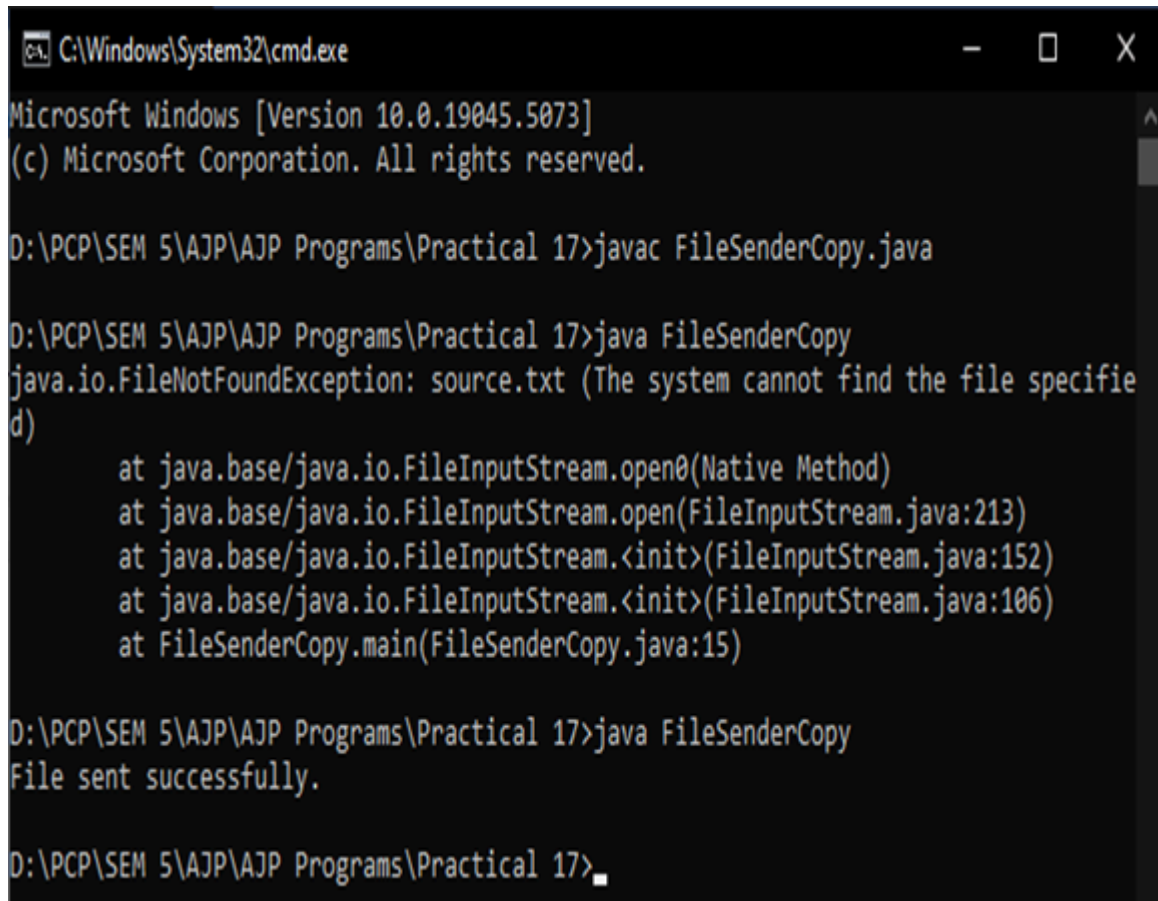
```
            fis.close();
```

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

```

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

```



The screenshot shows a Windows Command Prompt window titled "C:\Windows\System32\cmd.exe". The window displays the following commands and output:

```

Microsoft Windows [Version 10.0.19045.5073]
(c) Microsoft Corporation. All rights reserved.

D:\PCP\SEM 5\AJP\AJP Programs\Practical 17>javac FileSenderCopy.java

D:\PCP\SEM 5\AJP\AJP Programs\Practical 17>java FileSenderCopy
java.io.FileNotFoundException: source.txt (The system cannot find the file specifie
d)
    at java.base/java.io.FileInputStream.open0(Native Method)
    at java.base/java.io.FileInputStream.open(FileInputStream.java:213)
    at java.base/java.io.FileInputStream.<init>(FileInputStream.java:152)
    at java.base/java.io.FileInputStream.<init>(FileInputStream.java:106)
    at FileSenderCopy.main(FileSenderCopy.java:15)

D:\PCP\SEM 5\AJP\AJP Programs\Practical 17>java FileSenderCopy
File sent successfully.

D:\PCP\SEM 5\AJP\AJP Programs\Practical 17>

```

2. FileRecieverCopy.java

```

import java.io.FileOutputStream;

import java.io.IOException;

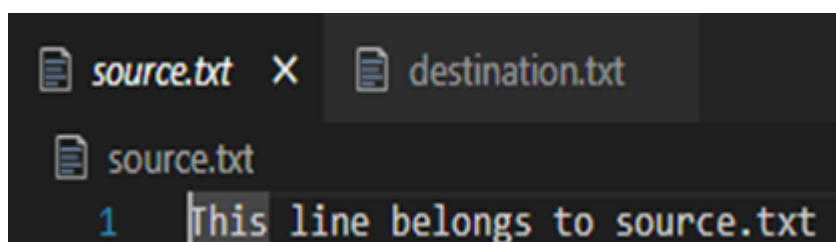
import java.net.DatagramPacket;

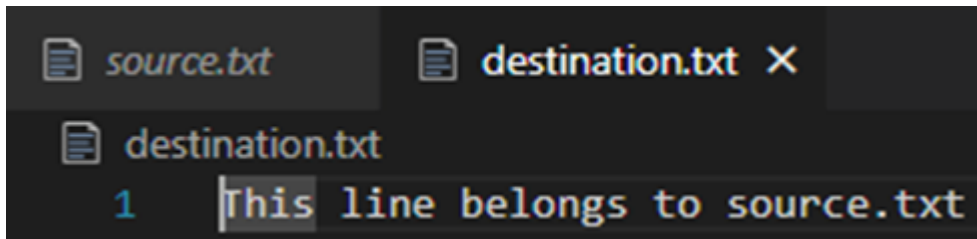
import java.net.DatagramSocket;

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();

    }

}
}
}

```

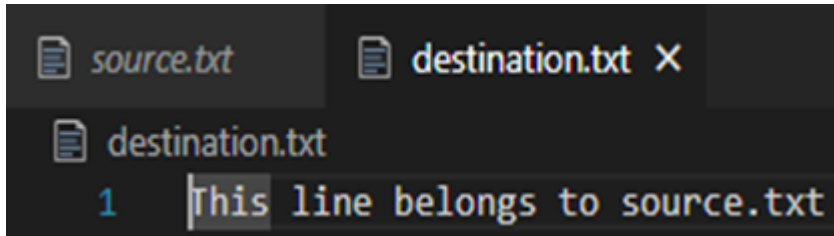
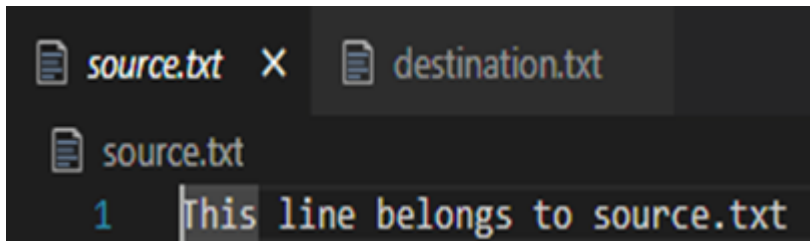
The image shows two side-by-side terminal windows, both titled 'C:\Windows\System32\cmd.exe'. The left window shows the execution of 'FileRecieverTransfer.java' (note the typo in the filename). The output shows 'Receiving file contents...' and 'File received and saved to destination.txt.'. The right window shows the execution of 'FileSenderTransfer.java'. The output shows 'Sending file contents...' and 'File transfer completed.'. Both windows show the directory 'D:\PCP\SEM 5\AJP\AJP Programs\Practical 17' as the current directory.

```

C:\Windows\System32\cmd.exe
D:\PCP\SEM 5\AJP\AJP Programs\Practical 17>javac FileRecieverTransfer.java
D:\PCP\SEM 5\AJP\AJP Programs\Practical 17>java FileRecieverTransfer
Receiving file contents...
File received and saved to destination.txt.
D:\PCP\SEM 5\AJP\AJP Programs\Practical 17>

C:\Windows\System32\cmd.exe
D:\PCP\SEM 5\AJP\AJP Programs\Practical 17>javac FileSenderTransfer.java
D:\PCP\SEM 5\AJP\AJP Programs\Practical 17>java FileSenderTransfer
Sending file contents...
File transfer completed.
D:\PCP\SEM 5\AJP\AJP Programs\Practical 17>

```



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();  
}  
}  
}
```