# Lab 2

1.)Write a Program to implement file transfer between two systems in the network.

FileServer.java

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

public class FileServer extends Thread {

    private ServerSocket ss;    //create server socket

    public FileServer(int port) {
        try {
            ss = new ServerSocket(port);  //initialize socket on given port
        } catch (IOException e) {
            e.printStackTrace();
        }
    }

    public void run() {
        while (true) {
            try {
                Socket clientSock = ss.accept(); //accept client connection
                saveFile(clientSock);            //invoke method for saving file
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }

    private void saveFile(Socket clientSock) throws IOException {
        DataInputStream dis = new DataInputStream(clientSock.getInputStream());
        FileOutputStream fos = new FileOutputStream("servercopy.py");
        byte[] buffer = new byte[4096];

        int filesize = 15123; // Send file size in separate msg
        int read = 0;
        int totalRead = 0;
        int remaining = filesize;
        while((read = dis.read(buffer, 0, Math.min(buffer.length, remaining))) > 0) {
//read file from client
            totalRead += read;
            remaining -= read;
```

```
            System.out.println("read " + totalRead + " bytes.");
            fos.write(buffer, 0, read);                         //write using file output
stream
        }

        fos.close();
        dis.close();
    }

    public static void main(String[] args) {
        FileServer fs = new FileServer(1988);     //create new object fs
        fs.start();
    }

}
```

FileClient.java

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

public class FileClient {

    private Socket s;

    public FileClient(String host, int port, String file) {      //initialize with
localhost and port 1988 and file server.py
        try {
            s = new Socket(host, port);
            sendFile(file);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    public void sendFile(String file) throws IOException {
        DataOutputStream dos = new DataOutputStream(s.getOutputStream());
        FileInputStream fis = new FileInputStream(file);
        byte[] buffer = new byte[4096];

        while (fis.read(buffer) > 0) {
            dos.write(buffer);
        }

        fis.close();
```

```
        dos.close();
    }

    public static void main(String[] args) {
        FileClient fc = new FileClient("localhost", 1988, "server.py");
    }

}
```

2. .)Write a program for remote method invocation

Interface Class

```
import java.rmi.Remote;
import java.rmi.RemoteException;

public interface ConnectInterface extends Remote{
   public String sayThis(String th) throws RemoteException;
}
```

Method Class

```
import java.rmi.*;
import java.rmi.server.*;
class MethodClass extends UnicastRemoteObject implements ConnectInterface{
   MethodClass()throws RemoteException{
       super();
       }
       public String sayThis(String th){
           th=" Hello World";
           return th;
       }
}
```

RmiServer

```
import java.rmi.*;
import java.rmi.registry.*;
class RmiServer{
    public static void main(String[] args) {
       try{
           ConnectInterface stubt=new MethodClass();
           Naming.rebind("rmi://localhost:5000/sonoo",stubt);
           //System.out.println(stub.sayThis("Hello World"));
           }catch(Exception e){}
   }
}
```

RMIClient

```
import java.rmi.*;
class RmiClient{
   public static void main(String[] arg){
       try{
           ConnectInterface stub=(ConnectInterface)
Naming.lookup("rmi://localhost:5000/sonoo");
           System.out.println(stub.sayThis("Hello World"));
           }catch(Exception e){}


   }

}
```