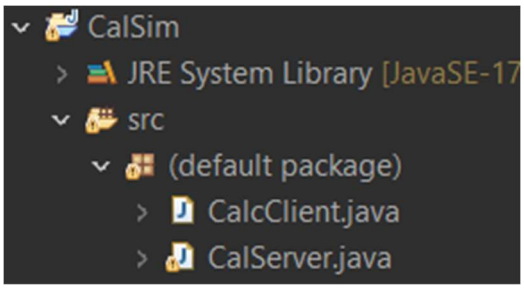
Practical No. 2

Implementation of Remote Procedure Call

Q.1 . Write a java program to implement a Server calculator using RPC concept. (Make use of datagram).

Structure :



Program :

CalcClient.java

```
import java.io.BufferedReader;
import java.io.DataInputStream;
import java.io.DataOutputStream;
import java.io.InputStreamReader;
import java.net.Socket;
public class CalcClient {
	Socket socket;
	int port;

	public CalcClient(int port) {
		this.port=port;
	}

	public void sndReq() throws Exception{
		socket=new Socket("localhost",port);
```

```
DataInputStream din=new DataInputStream(socket.getInputStream());
DataOutputStream dout=new DataOutputStream(socket.getOutputStream());
BufferedReader in=new BufferedReader(new InputStreamReader(System.in));

String str="";
int num1,num2;

System.out.println("1:Addition \n2:Sub \n3:Multi \n4:Div \n5:Exit");
System.out.println("Enter your choice ");
int choice=Integer.parseInt(in.readLine());
System.out.println("Val=" +choice);

switch(choice) {
case 1:
        str += choice+"-";
        System.out.println("Enter 1st Number\n");
        num1 =Integer.parseInt(in.readLine());
        str +=num1+"-";
        System.out.println("Enter 2nd Number\n");
        num2 =Integer.parseInt(in.readLine());
        str +=num2+"-";
        break;

case 2:
        str += choice+"-";
        System.out.println("Enter 1st Number");
        num1 =Integer.parseInt(in.readLine());
        str +=num1+"-";
```

```
            System.out.println("Enter 2nd Number");
            num2 =Integer.parseInt(in.readLine());
            str +=num2+"-";
            break;

case 3:
            str += choice+"-";
            System.out.println("Enter 1st Number");
            num1 =Integer.parseInt(in.readLine());
            str +=num1+"-";
            System.out.println("Enter 2nd Number");
            num2 =Integer.parseInt(in.readLine());
            str +=num2+"-";
            break;

case 4:
            str += choice+"-";
            System.out.println("Enter 1st Number");
            num1 =Integer.parseInt(in.readLine());
            str +=num1+"-";
            System.out.println("Enter 2nd Number");
            num2 =Integer.parseInt(in.readLine());
            str +=num2+"-";
            break;

case 5:
            System.out.println("Program Exited!");
            break;
```

```
                default:
                        System.out.println("Invalid option!");
                        break;
                }
                System.out.println(str);
                dout.writeUTF(str);
                dout.flush();
                String result=din.readUTF();
                System.out.println("Result is"+result);
                din.close();
                dout.close();
                socket.close();
        }
        public static void main(String[] args) {
                // TODO Auto-generated method stub
                try {
                        CalcClient cc=new CalcClient(5000);
                        cc.sndReq();
                }
                catch (Exception e) {
                        // TODO: handle exception
                        System.out.println(e.getMessage());
                }
        }
}
```

CalcServer.java

```
import java.io.DataInputStream;
import java.io.DataOutputStream;
```

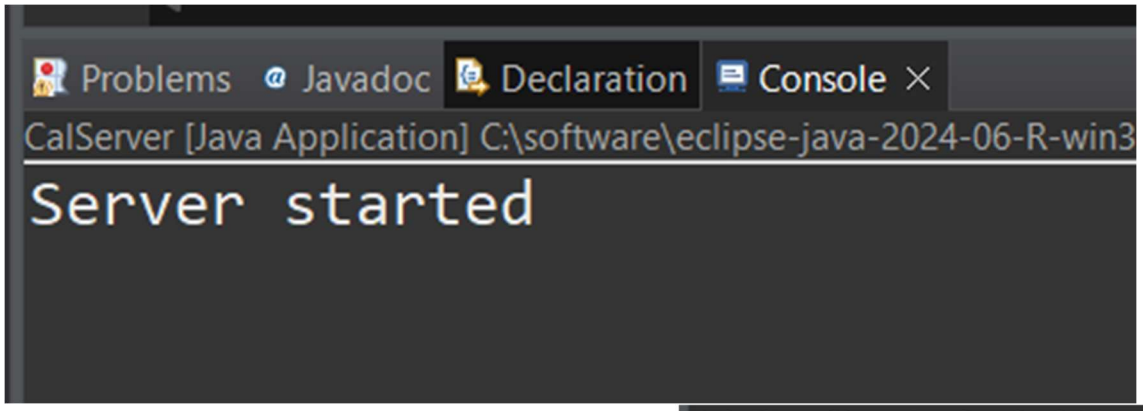
```
import java.net.ServerSocket;
import java.net.Socket;
import java.security.PublicKey;
import java.util.StringTokenizer;

public class CalServer {
	int port;
	ServerSocket ss;
	Socket socket;

	public CalServer() {
		this.port=0;
	}

	public CalServer(int port) {
		this.port=port;
	}

	public double addition(int n1,int n2) {
		return n1+n2;
	}

	public double sub(int n1,int n2) {
		return n1-n2;
	}

	public double mul(int n1,int n2) {
		return n1*n2;
```

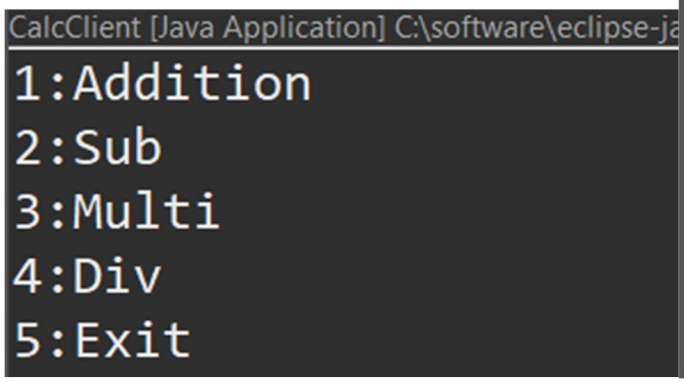
```
	}

	public double div(int n1,int n2) {
		return n1/n2;
	}

	public void listen() {
		try {
			System.out.println("Server started\n");
			ss=new ServerSocket(port);
			socket=ss.accept();

			DataInputStream dis=new DataInputStream(socket.getInputStream());
			DataOutputStream dout=new
DataOutputStream(socket.getOutputStream());
			double result=0.0;

			while(true) {
				String str=dis.readUTF();
				StringTokenizer st=new StringTokenizer(str,"-");
				int choice=Integer.parseInt(st.nextToken());
				int num1=Integer.parseInt(st.nextToken());
				int num2=Integer.parseInt(st.nextToken());
				CalServer cs=new CalServer();

				switch (choice) {
				case 1:
					result=cs.addition(num1, num2);
					break;
```

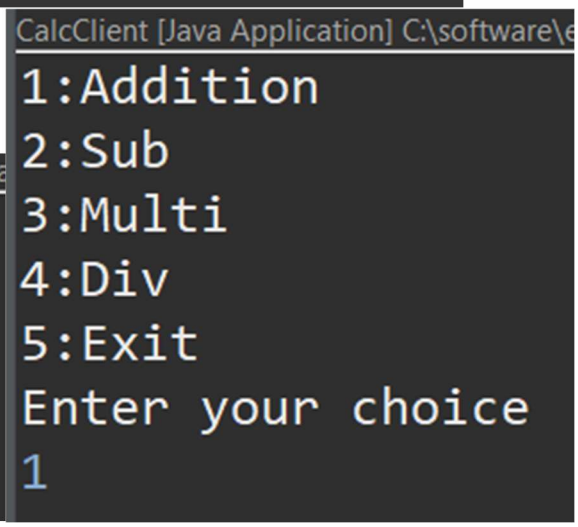
```
                case 2:
                        result=cs.sub(num1, num2);
                        break;

                case 3:
                        result=cs.mul(num1, num2);
                        break;

                case 4:
                        result=cs.div(num1, num2);
                        break;
                }

                System.out.println("Result for " +str+" is - ");
                String res=Double.toString(result);
                System.out.print(res);
                dout.writeUTF(res);
                dout.flush();
                dis.close();
                dout.close();
                socket.close();

                }

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

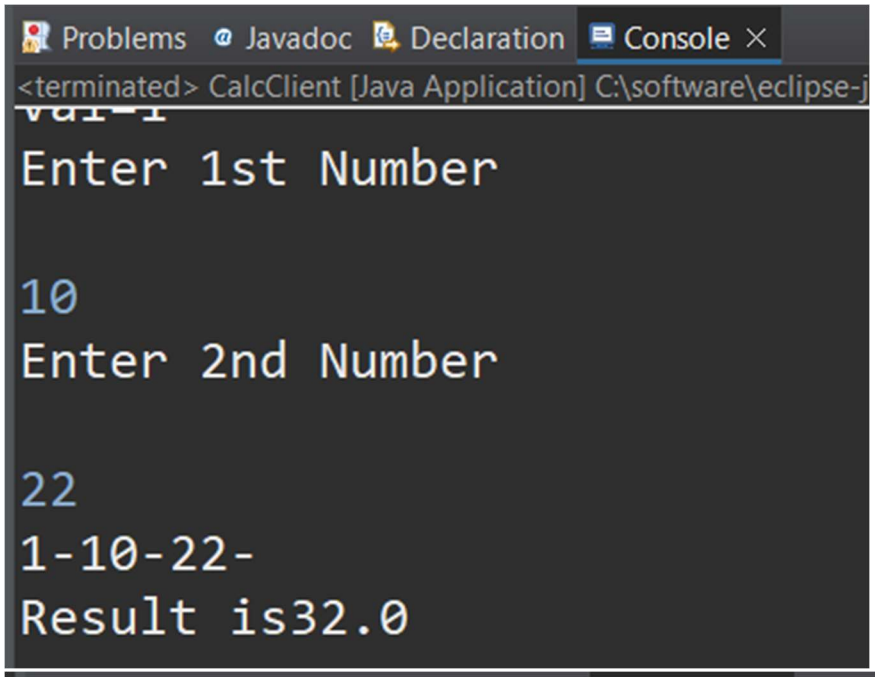
```
                }


        }
        public static void main(String arg[]) {
                CalServer cc=new CalServer(5000);
                cc.listen();
        }
}
```

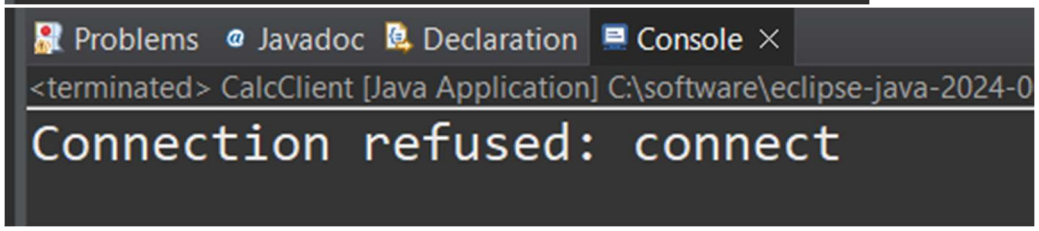
Output :





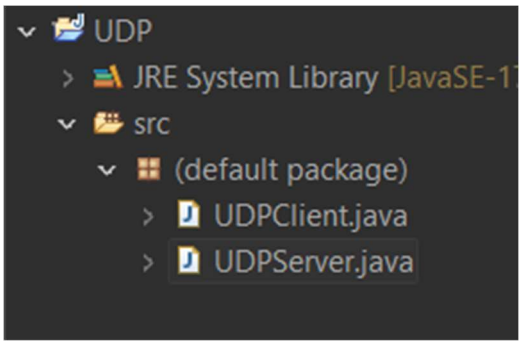



Problems
Javadoc
Declaration
Console
<terminated> CalcClient [Java Application] C:\software\eclipse-j
Enter 1st Number
10
Enter 2nd Number
22
1-10-22-
Result is32.0


Problems
Javadoc
Declaration
Console
<terminated> CalcClient [Java Application] C:\software\eclipse-java-2024-0
Connection refused: connect

Q.2 Write a java to implement a Date Time Server using RPC concept. (Make use of datagram).

Structure :



Program :

UDPClient.java

```
import java.io.IOException;
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.SocketException;


public class UDPClient {


	public static void main(String[] args) throws SocketException {
		// TODO Auto-generated method stub
		DatagramPacket dpac;
		DatagramSocket dsoc = new DatagramSocket(1314);
		byte[] b = new byte[64];
		String data = "No Data";
		System.out.println("Client up");
		try {
			while(true) {
				dpac = new DatagramPacket(b, b.length);
				dsoc.receive(dpac);
```

```
                    data = new String(dpac.getData());
                    System.out.println("We received Data : " + data);
                }
            }catch (IOException e) {
                // TODO: handle exception
                System.out.println("IOException");
            }
            dsoc.close();
        }
}
```

UDPServer.java

```
import java.io.IOException;
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;
import java.net.SocketException;
import java.util.Date;

public class UDPServer {

    public static void main(String[] args) throws SocketException {
        // TODO Auto-generated method stub
        DatagramPacket dpac;
        DatagramSocket dsac = new DatagramSocket();
        System.out.println("Server up");
        try {
            while(true) {
                System.out.println("Sending");
```

```
                    Thread.sleep(1000);
                    String time = new Date().toString();
                    byte b[] = time.getBytes();
                    dpac = new DatagramPacket(b, b.length,
InetAddress.getByName("localhost"), 1314);
                    dsac.send(dpac);
                }
            } catch (IOException | InterruptedException e) {
                // TODO: handle exception
                System.out.println("IOException");
            }
            dsac.close();
        }
}
```

Output :