

Practical No. 02

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

Program: Server.java package rpc;

```
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;
import java.util.StringTokenizer;

public class Server { private
DatagramSocket udpSocket;
    private int port;

    public Server(int port) {
this.port = port;
    }
    public static int addition(int num1,int num2)
    {
        return num1+num2;
    }
    public static int subtraction(int num1,int num2)
    {
        return num1-num2;
    }
    public static int multiplication(int num1,int num2)
    {
        return num1*num2;
    }
    public static int division(int num1,int num2)
    {
        return num1/num2;
    }
}
```

```
private void listen() {
try {
    DatagramSocket udpSocket = new DatagramSocket(port);
    System.out.println("Server started at " + InetAddress.getLocalHost());
    String msg;

    byte[] buf = new byte[1024];
    DatagramPacket packet = new DatagramPacket(buf, buf.length);

    // blocks until a packet is received
    udpSocket.receive(packet);          msg = new
    String(packet.getData()).trim();

    StringTokenizer str=new StringTokenizer(msg,"-");
    int mthNo=Integer.parseInt(str.nextToken());
    int num1=Integer.parseInt(str.nextToken());
    int num2=Integer.parseInt(str.nextToken());
    int result;          if(mthNo==1)
    {
        result=addition(num1,num2);
        msg="Addition:"+result;
    }
    if(mthNo==2)
    {
        result=substraction(num1,num2);
        msg="substraction:"+result;
    }
    if(mthNo==3)
    {
        result=multiplication(num1,num2);
        msg="multiplication:"+result;
```

```
        }
    if(mthNo==4)
    {
        result=division(num1,num2);
        msg="division:"+result;
    }

    System.out.println("Message from " + packet.getAddress().getHostAddress() +
": " + msg);

}

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

finally {
    //udpSocket.close();
}

}

public static void main(String[] args) {
    Server client = new Server(5000);
    client.listen();
}

}
```

Client.java package

```
rpc;
```

```
import java.io.BufferedReader; import
java.io.InputStreamReader; import
java.net.DatagramPacket; import
java.net.DatagramSocket; import
```

```
java.net.InetAddress; import
```

```
java.util.Scanner;
```

```
public class Client {
```

```
    DatagramSocket udpSocket;
```

```
    InetAddress serverAddress;
```

```
    int port;
```

```
    Scanner scanner;
```

```
    public Client(int port) {
```

```
        this.port = port;
```

```
    }
```

```
    public void sendReq() {
```

```
        String in;
```

```
        try {
```

```
            udpSocket = new DatagramSocket();
```

```
            InetAddress host = InetAddress.getLocalHost();    serverAddress
```

```
            = InetAddress.getByName(host.getHostName());
```

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

```
            System.out.println("UDP Client started at " + InetAddress.getLocalHost());
```

```
            String paramlist="";
```

```
            System.out.println("Enter  
Method:\n1.Addition:\n2.Subtraction\n3.Multiplication\n4.Devision");
```

```
            in = keyRead.readLine();    paramlist=paramlist+in+"-";
```

```
            System.out.println("Enter Number 1:");
```

```
            in = keyRead.readLine();
```

```
            paramlist=paramlist+in+"-";
```

```
        System.out.println("Enter Number 2:");

        in = keyRead.readLine();

        paramlist=paramlist+in;

        DatagramPacket p = new DatagramPacket(paramlist.getBytes(),
        paramlist.getBytes().length, serverAddress, port);        udpSocket.send(p);

    }

    catch(Exception e) {

        System.out.println(e.getMessage());

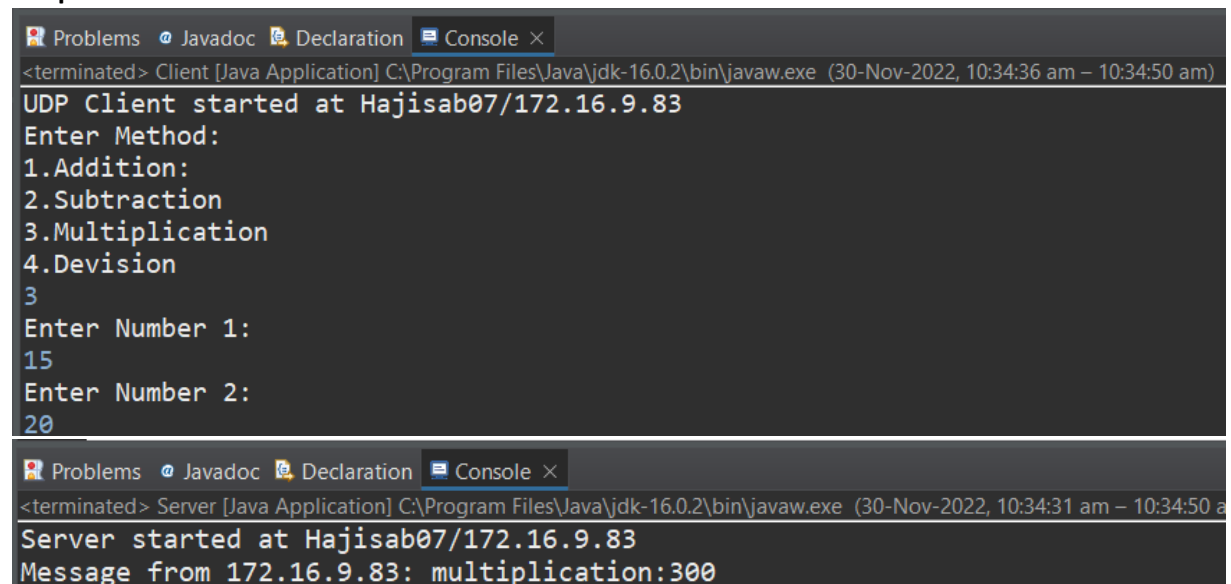
    }

}

public static void main(String[] args) {
Client sender = new Client(5000);
sender.sendReq();

}

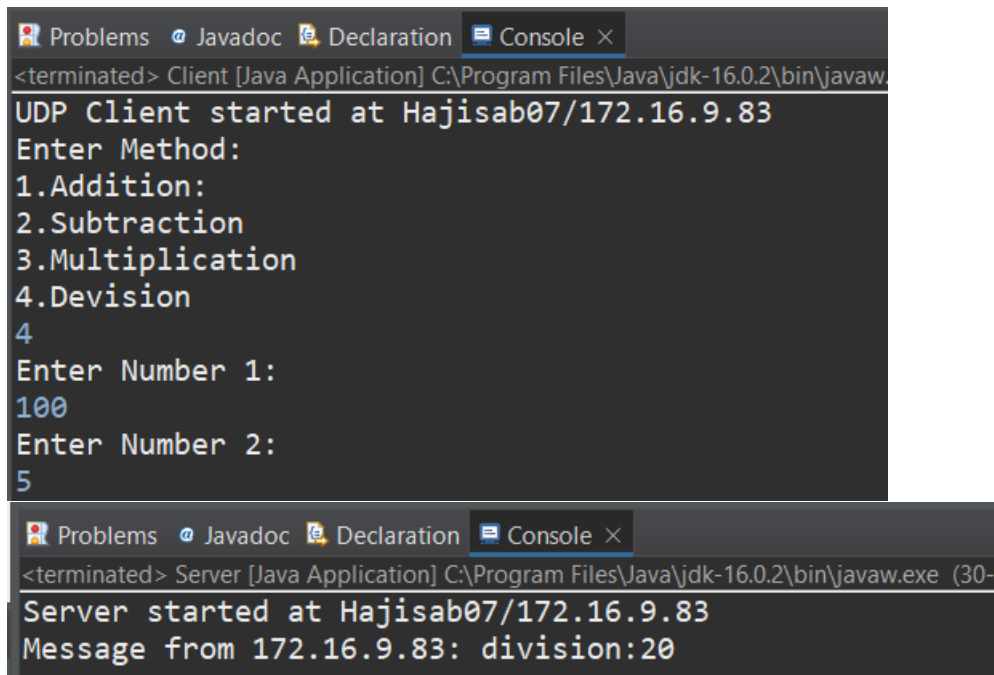
}
```

Output:

The image shows two screenshots of a Java IDE's console window. The top screenshot shows the output of the 'Client' application. It starts with the message '<terminated> Client [Java Application] C:\Program Files\Java\jdk-16.0.2\bin\javaw.exe (30-Nov-2022, 10:34:36 am – 10:34:50 am)'. The client then prints 'UDP Client started at Hajisab07/172.16.9.83'. It prompts the user to 'Enter Method:' and lists four options: '1.Addition:', '2.Subtraction', '3.Multiplication', and '4.Devision'. The user enters '3'. It then prompts 'Enter Number 1:' and the user enters '15'. Finally, it prompts 'Enter Number 2:' and the user enters '20'. The bottom screenshot shows the output of the 'Server' application. It starts with the message '<terminated> Server [Java Application] C:\Program Files\Java\jdk-16.0.2\bin\javaw.exe (30-Nov-2022, 10:34:31 am – 10:34:50 a)'. The server prints 'Server started at Hajisab07/172.16.9.83'. It then receives a message from the client and prints 'Message from 172.16.9.83: multiplication:300'.

```
Problems Javadoc Declaration Console X
<terminated> Client [Java Application] C:\Program Files\Java\jdk-16.0.2\bin\javaw.exe (30-Nov-2022, 10:34:36 am – 10:34:50 am)
UDP Client started at Hajisab07/172.16.9.83
Enter Method:
1.Addition:
2.Subtraction
3.Multiplication
4.Devision
3
Enter Number 1:
15
Enter Number 2:
20

Problems Javadoc Declaration Console X
<terminated> Server [Java Application] C:\Program Files\Java\jdk-16.0.2\bin\javaw.exe (30-Nov-2022, 10:34:31 am – 10:34:50 a)
Server started at Hajisab07/172.16.9.83
Message from 172.16.9.83: multiplication:300
```



The image shows two screenshots of a Java IDE console. The top screenshot shows the output of a UDP client program. It starts with the message "<terminated> Client [Java Application] C:\Program Files\Java\jdk-16.0.2\bin\javaw.exe". The client is started at Hajisab07/172.16.9.83. It prompts the user to "Enter Method:" and lists four options: 1.Addition, 2.Subtraction, 3.Multiplication, and 4.Devision. The user enters "4". It then prompts for "Enter Number 1:" and the user enters "100". It then prompts for "Enter Number 2:" and the user enters "5". The bottom screenshot shows the output of a server program. It starts with the message "<terminated> Server [Java Application] C:\Program Files\Java\jdk-16.0.2\bin\javaw.exe (30-1)". The server is started at Hajisab07/172.16.9.83. It receives a message from 172.16.9.83: division:20.

```
<terminated> Client [Java Application] C:\Program Files\Java\jdk-16.0.2\bin\javaw.exe
UDP Client started at Hajisab07/172.16.9.83
Enter Method:
1.Addition:
2.Subtraction
3.Multiplication
4.Devision
4
Enter Number 1:
100
Enter Number 2:
5

<terminated> Server [Java Application] C:\Program Files\Java\jdk-16.0.2\bin\javaw.exe (30-1)
Server started at Hajisab07/172.16.9.83
Message from 172.16.9.83: division:20
```

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

Program:

Server.java package

rpcdatetime;

```
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress; import
java.time.LocalDateTime;
```

```
public class Server {
    private DatagramSocket udpSocket;
    private int port;

    public Server(int port) {
this.port = port;
    }
    public static LocalDateTime date()
    {
```

```
        return java.time.LocalDateTime.now();
    }

    private void listen() {
    try {

        DatagramSocket udpSocket = new DatagramSocket(port);
        System.out.println("Server started at " + InetAddress.getLocalHost());
        LocalDateTime msg;

        byte[] buf = new byte[1024];
        DatagramPacket packet = new DatagramPacket(buf, buf.length);

        // blocks until a packet is received
        udpSocket.receive(packet);

        msg=date();
        System.out.println("Message from " + packet.getAddress().getHostAddress() + ": " +
        msg);

    }
    catch(Exception e) {
        System.out.println(e.getMessage());
    }
    finally {
        //udpSocket.close();
    }
    }

    public static void main(String[] args) {
    Server client = new Server(5000);
    client.listen();
```

```
}  
}
```

Client.java package

```
rpcdatetime;
```

```
import java.io.BufferedReader;  
import java.io.InputStreamReader;  
import java.net.DatagramPacket;  
import java.net.DatagramSocket;  
import java.net.InetAddress; import  
java.time.LocalDateTime; import  
java.util.Scanner;
```

```
public class Client {  
    DatagramSocket udpSocket;  
    InetAddress serverAddress;  
    int port;  
    Scanner scanner;
```

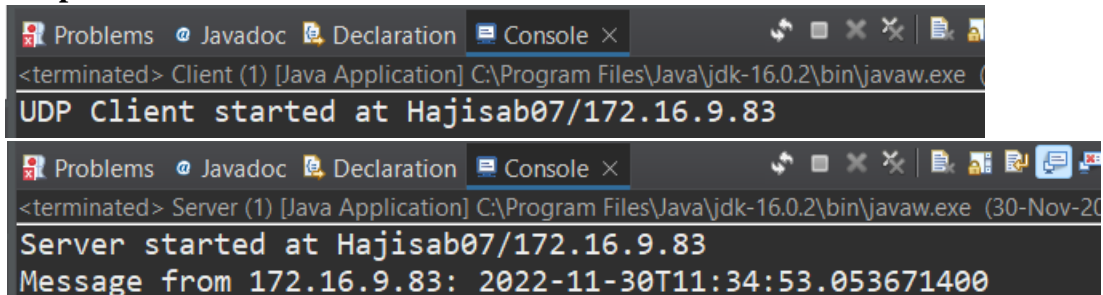
```
    public Client(int port) {  
        this.port = port;  
    }
```

```
    public void sendReq() {  
        String in;    try {  
            udpSocket = new DatagramSocket();  
            InetAddress host = InetAddress.getLocalHost();  
            serverAddress = InetAddress.getByName(host.getHostName());
```

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



```
        System.out.println("UDP Client started at " + InetAddress.getLocalHost());  
        String paramlist="";  
        DatagramPacket p      =      new      DatagramPacket(paramlist.getBytes(),  
paramlist.getBytes().length, serverAddress, port);      udpSocket.send(p);  
  
    }  
    catch(Exception e) {  
        System.out.println(e.getMessage());  
    }  
}  
  
public static void main(String[] args) {  
    Client sender = new Client(5000);  
    sender.sendReq();  
}  
  
}
```

Output:

The image shows two screenshots of a Java IDE console. The top screenshot shows the output of a client application: "<terminated> Client (1) [Java Application] C:\Program Files\Java\jdk-16.0.2\bin\javaw.exe (UDP Client started at Hajisab07/172.16.9.83". The bottom screenshot shows the output of a server application: "<terminated> Server (1) [Java Application] C:\Program Files\Java\jdk-16.0.2\bin\javaw.exe (30-Nov-20 Server started at Hajisab07/172.16.9.83 Message from 172.16.9.83: 2022-11-30T11:34:53.053671400".