201351010
Aditya Prakash
DC LAB-06

QUESTION 1:

ATMInterface

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

public interface ATMInterface extends Remote
{
	public void deposit(int account, double amount) throws RemoteException;
	public void withdraw(int account, double amount) throws RemoteException;
	public double inquiry(int account) throws RemoteException;
	public void CreateAcc() throws RemoteException;
}
```

ATMImpl

```
import java.rmi.*;
import java.rmi.server.*;
import java.util.*;

public class ATMImpl extends UnicastRemoteObject implements ATMInterface
{
	private HashMap<Integer,Double> h = new HashMap<Integer,Double>();

	public ATMImpl() throws RemoteException
	{
		super();
	}

	public void CreateAcc()
	{
		h.put(100,400.0);
		h.put(1,300.0);
		h.put(2,200.0);
		h.put(3,100.0);
	}

	public void deposit(int account, double amount) throws RemoteException
	{
		h.put(account,h.get(account)+amount);
	}
```

```
	public void withdraw(int account,double amount) throws RemoteException
	{
		if(h.get(account)>0)
		{
		   h.put(account,h.get(account)-amount);
		}
	}

	public double inquiry(int account) throws RemoteException
	{
		return h.get(account);
	}
}
```

Client

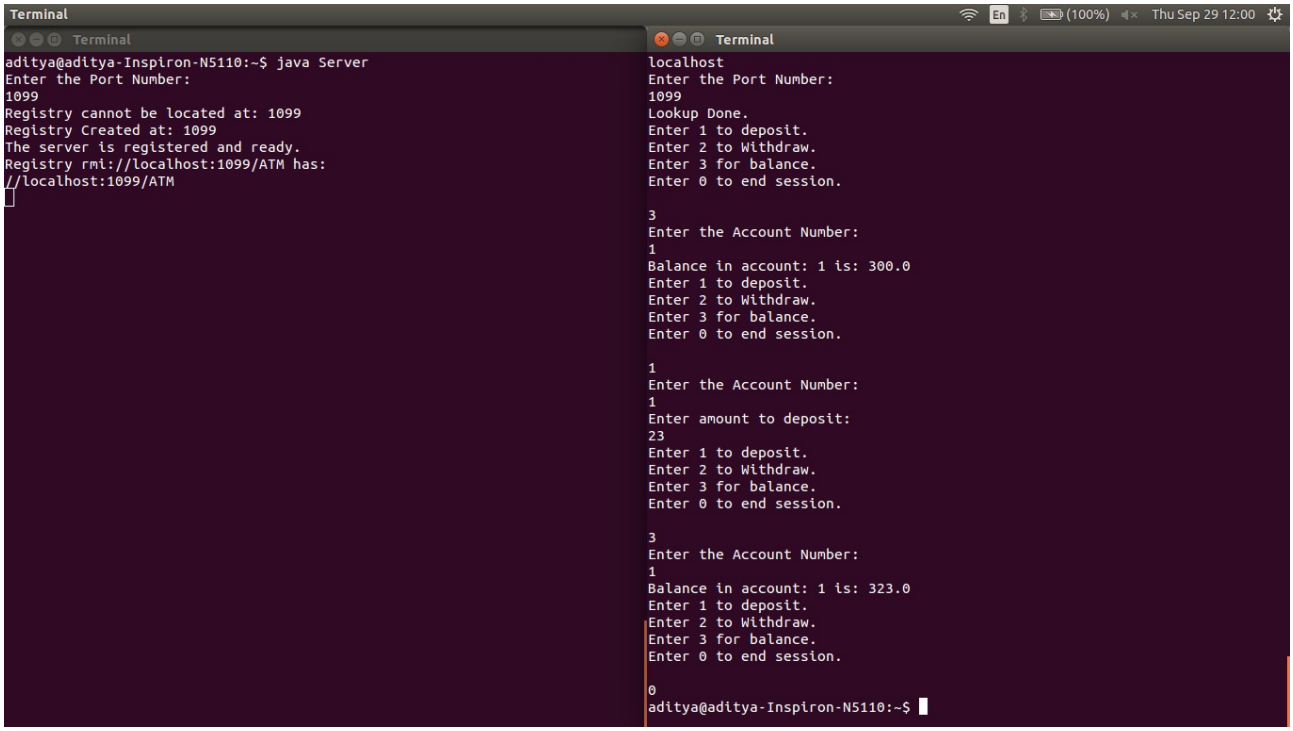
```
import java.io.*;
import java.rmi.Naming;
import java.util.*;

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

		try
		{
			int RMIPort;
			String hostName;
			InputStreamReader is = new InputStreamReader(System.in);
			BufferedReader br = new BufferedReader(is);
			System.out.println("Enter the Host Name: ");
			hostName = br.readLine();
			System.out.println("Enter the Port Number: ");
			String portNum = br.readLine();
			RMIPort = Integer.parseInt(portNum);
			String registryURL = "rmi://" + hostName + ":" + RMIPort + "/ATM";

			ATMInterface h = (ATMInterface)Naming.lookup(registryURL);
		   System.out.println("Lookup Done.");
		   h.CreateAcc();
		   Scanner s = new Scanner(new InputStreamReader(System.in));

		   int num=1,acc;
		   double amt;
		   while(num !=0)
		   {
				System.out.println("Enter 1 to deposit.\nEnter 2 to Withdraw.\nEnter 3
for balance.\nEnter 0 to end session.\n");
				num = s.nextInt();
				switch(num)
				{
```

```
                                        case 1:
                                                System.out.println("Enter the Account Number:
");
                                                acc = s.nextInt();
                                                System.out.println("Enter amount to deposit: ");
                                                amt = s.nextDouble();
                                                h.deposit(acc,amt);
                                                break;

                                        case 2:
                                                System.out.println("Enter the Account Number:
");
                                                acc = s.nextInt();
                                                System.out.println("Enter the amount to
withdraw: ");
                                                amt = s.nextDouble();
                                                h.withdraw(acc, amt);
                                                break;

                                        case 3:
                                                System.out.println("Enter the Account Number:
");
                                                acc = s.nextInt();
                                                System.out.println("Balance in account: " + acc
+ " is: " + h.inquiry(acc));
                                                break;

                                        case 0:
                                                num = 0;
                                                break;
                                }
                                }
                        }
                        catch(Exception e)
                        {
                                System.out.println("Exception: " + e);
                        }
        }
}
```

Server

```
import java.io.*;
import java.net.MalformedURLException;
import java.rmi.*;
import java.rmi.registry.LocateRegistry;
import java.rmi.registry.Registry;

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

```
                InputStreamReader is = new InputStreamReader(System.in);
                BufferedReader br = new BufferedReader(is);
                String portNum, registryURL, registryURL2;
                try
                {
                        System.out.println("Enter the Port Number:");
                        portNum = (br.readLine()).trim();
                        int RMIPortNum = Integer.parseInt(portNum);
                        startRegistry(RMIPortNum);

                        ATMImpl exportedObj = new ATMImpl();
                        registryURL = "rmi://localhost:" + portNum + "/ATM";
                        Naming.rebind(registryURL, exportedObj);
                        System.out.println("The server is registered and ready.");
                        listRegistry(registryURL);
                }
                catch(Exception e)
                {
                        System.out.println("Exception in HelloServer.main: " + e );
                }

        }

        private static void startRegistry(int RMIPortNum) throws RemoteException
        {
                try
                {
                        Registry reg = LocateRegistry.getRegistry(RMIPortNum);
        reg.list();
                }
                catch(RemoteException e)
                {
                        System.out.println("Registry cannot be located at: " + RMIPortNum);
                        LocateRegistry.createRegistry(RMIPortNum);
                        System.out.println("Registry Created at: " + RMIPortNum);
                }
        }
        private static void listRegistry(String registryURL) throws RemoteException,
MalformedURLException
        {
                System.out.println("Registry " + registryURL + " has: ");
                String[] n = Naming.list(registryURL);
                for(int i=0; i < n.length; i++)
                {
                        System.out.println(n[i]);
                }
        }
}
```


Terminal
(100%)
Thu Sep 29 12:00
Terminal
aditya@aditya-Inspiron-N5110:~$ java Server
Enter the Port Number:
1099
Registry cannot be located at: 1099
Registry Created at: 1099
The server is registered and ready.
Registry rmi://localhost:1099/ATM has:
//localhost:1099/ATM
Terminal
localhost
Enter the Port Number:
1099
Lookup Done.
Enter 1 to deposit.
Enter 2 to Withdraw.
Enter 3 for balance.
Enter 0 to end session.
3
Enter the Account Number:
1
Balance in account: 1 is: 300.0
Enter 1 to deposit.
Enter 2 to Withdraw.
Enter 3 for balance.
Enter 0 to end session.
1
Enter the Account Number:
1
Enter amount to deposit:
23
Enter 1 to deposit.
Enter 2 to Withdraw.
Enter 3 for balance.
Enter 0 to end session.
3
Enter the Account Number:
1
Balance in account: 1 is: 323.0
Enter 1 to deposit.
Enter 2 to Withdraw.
Enter 3 for balance.
Enter 0 to end session.
0
aditya@aditya-Inspiron-N5110:~$

Question 2:

Client

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

public class Cli
{
	public static void main(String[] args) throws Exception
	{
		Socket client = new Socket("localhost",1245);
		double[] data = new double[5];

		data[0] = 99;
		data[1] = 88;
		data[2] = 77;
		data[3] = 66;
		data[4] = 55;

		DataOutputStream Out = new DataOutputStream(client.getOutputStream());
		Out.writeInt(data.length);
		for(int i=0;i<data.length;i++)
		{
			Out.writeDouble(data[i]);
	   }

		DataInputStream In = new DataInputStream(client.getInputStream());
		int len = In.readInt();
		for(int i=0;i<len;i++)
		{
			data[i] = In.readDouble();
			System.out.println(data[i] + "\n");
		}
	}
}
```

Server

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

public class Ser
{
	public static void main(String[] args) throws Exception
	{
		ServerSocket Server = new ServerSocket(1245);
		System.out.println("The server has started.");
		Socket client = Server.accept();
      byte[] data = null;
      DataInputStream In = new DataInputStream(client.getInputStream());
		int len = In.readInt();
      double[] arr = new double[len];
      for(int i=0;i<len;i++)
```

```
        {
           arr[i] = In.readDouble();
           System.out.println(arr[i]+"\n");
           arr[i] = Math.sqrt(arr[i]);
        }
                DataOutputStream Out = new DataOutputStream(client.getOutputStream());
                Out.writeInt(arr.length);
                for(int i=0;i<len;i++)
                {
                        Out.writeDouble(arr[i]);
                }
        }
}
```