

## ResultRMI.java

```
public interface ResultRMI extends java.rmi.Remote
{
    String calcresult(double marks[],int count) throws java.rmi.RemoteException;
}
```

## Create Interface

## ResultRMIImplementation

```
// ResultRMIImpl.java, ResultRMI implementation

import java.rmi.*;
import java.rmi.server.UnicastRemoteObject;

public class ResultRMIImpl extends UnicastRemoteObject implements ResultRMI
{
    public ResultRMIImpl(String name) throws RemoteException
    {
        super();
        try
        {
            Naming.rebind(name, this);
        }
        catch (Exception e)
        {
            System.out.println("Exception: " + e.getMessage());
            e.printStackTrace();
        }
    }

    public String calcresult(double marks[],int count) throws RemoteException
    {
        double total=0.0,average;
        for(int i=0;i<count;i++)
        {
            total=total+marks[i];
        }
        average = total/count;
    }
}
```

```

        if(average >= 70)
        {
            return"Distinction";
        }
        else if(average >=60 && average < 70)
        {
            return"Firstclass";
        }
        else if(average >=50 && average < 60)
        {
            return"secondclass";
        }
        else if(average >=40 && average <50)
        {
            return"passcalss";
        }
        else
        {
            return"fail";
        }
    }
}

```

Implement calcresult() and assign grade accordingly

ResultRMIServer.java

```

// ResultRMIServer.java

import java.rmi.*;
import java.rmi.server.*;

public class ResultRMIServer
{

    public static void main(String args[])
    {

        try
        {
            // Create ResultRMIImp1
            ResultRMIImp1 myResult = new ResultRMIImp1("rmi://localhost:5000/");
            System.out.println("ResultRMI Server ready.");
        }
    }
}

```

```

catch (Exception e)
{
    System.out.println("Exception: " + e.getMessage());
    e.printStackTrace();
}
}
}

// ResultRMIClient.java   RMI Result client

import java.rmi.*;
import java.rmi.registry.*;
import java.rmi.server.*;
import java.io.*;
import java.lang.*;

public class ResultRMIClient
{ public static void main(String args[]) throws Exception
{

    FileReader fr = new FileReader("studentdetails.txt");
    StreamTokenizer tok = new StreamTokenizer(fr);
    String name="",classval,s;
    double marks[] = new double[10];
    int i=0;
    tok.eolIsSignificant(true);
    try
    {
        ResultRMI myResult = (ResultRMI)Naming.lookup("rmi://localhost:5000/");

        while( tok.nextToken() != tok.TT_EOF)
        {

            if(tok.ttype == tok.TT_WORD)
            {
                name = tok.sval;
                System.out.println("\nname is:"+name);
                System.out.println("Obtained marks are:");
            }

            if(tok.ttype == tok.TT_NUMBER)
            {

                marks[i] = tok.nval;

```

```
        System.out.println("Marks of subject "+i+" is "+marks[i]);
        i++;

    }

    if(tok.ttype == tok.TT_EOL && tok.ttype != tok.TT_EOF)
    {
        classval = myResult.calcrestult(marks,i);
        System.out.println(name +" has " + classval);
        i=0;

    }

}

fr.close();

}

catch (Exception e)
{
    System.err.println("System Exception" + e);
}

System.exit(0);
}
}
```