PGM NO: 1

EMPLOYEE DETAILS

DATE:22/06/2016

AIM:

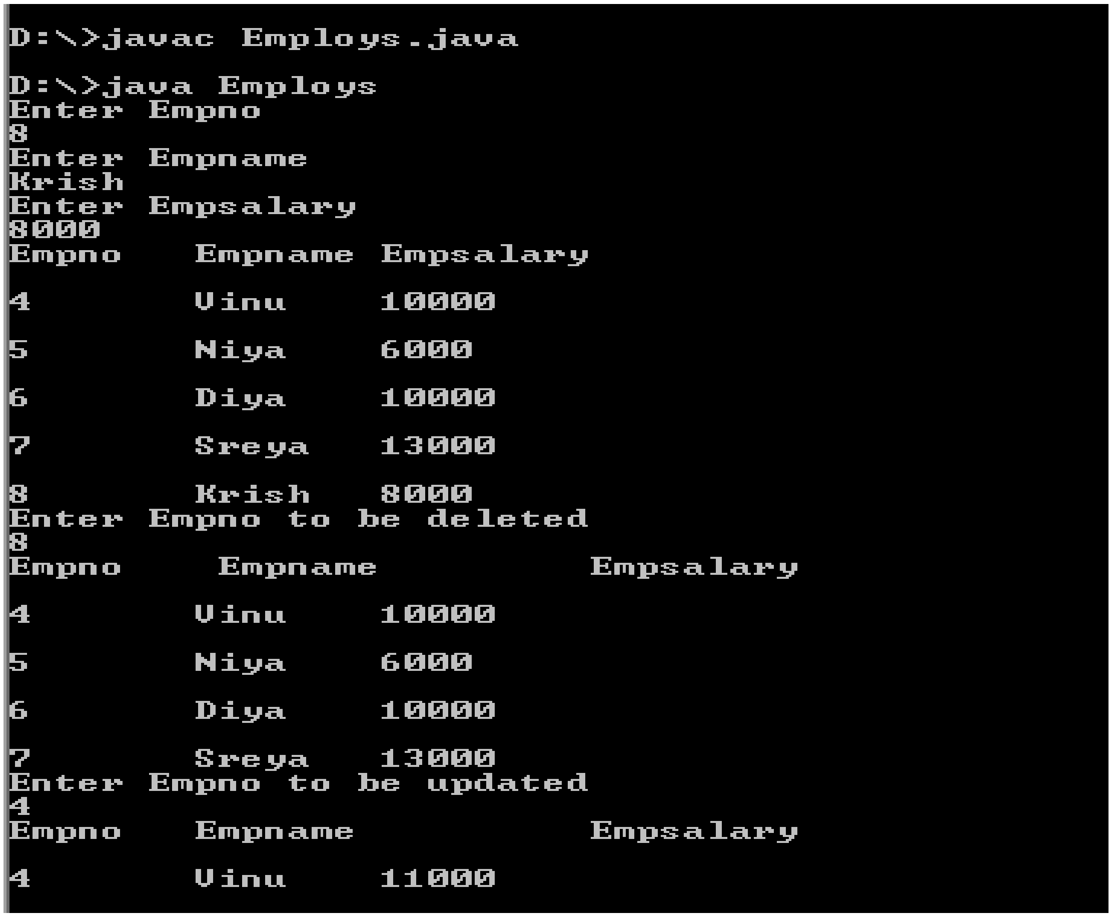
JDBC program to insert, Delete and Update records into Employee table.

PROGRAM:

```
import java.sql.*;

import java.io.*;

class Employs

{

        public static void main(String[] args)

        {

                try

                {

                        Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

                        Connection con=DriverManager.getConnection("Jdbc:Odbc:emp");

                        Statement st=con.createStatement();

                        BufferedReader br=new BufferedReader(new
                        InputStreamReader(System.in));

                        System.out.println("Enter Empno");

                        int Empno=Integer.parseInt(br.readLine());

                        System.out.println("Enter Empname");

                        String Empname=br.readLine();

                        System.out.println("Enter Empsalary");

                        int Empsalary=Integer.parseInt(br.readLine());

                        st.executeUpdate("insert into Employee
values("+Empno+",'"+Empname+"',"+Empsalary+");");

ResultSet rs=st.executeQuery("select * from Employee");
System.out.println("Empno\tEmpname\tEmpsalary");
while(rs.next())
{

        String r=rs.getString("Empno");
        String n=rs.getString("Empname");
        String s=rs.getString("Empsalary");
        System.out.println("\n"+r+"\t"+n+"\t"+s);
}
System.out.println("Enter Empno to be deleted");
int r1=Integer.parseInt(br.readLine());
st.executeUpdate("Delete from Employee where Empno="+r1+";");
ResultSet rs1=st.executeQuery("select * from Employee");
System.out.println("Empno \t Empname \t Empsalary");
while(rs1.next())
{

        String r2=rs1.getString("Empno");
        String n2=rs1.getString("Empname");
        String s2=rs1.getString("Empsalary");
        System.out.println("\n"+r2+"\t"+n2+"\t"+s2);
}
```

```
            System.out.println("Enter Empno to be updated");

            int r3=Integer.parseInt(br.readLine());

            st.executeUpdate("update Employee set Empsalary=Empsalary"+"
            +"+1000+"  where Empno="+r3+";");

            ResultSet rs4=st.executeQuery("select * from Employee");

            System.out.println("Empno \tEmpname \t Empsalary");

            while(rs4.next())

            {


                String r4=rs4.getString("Empno");

                String n4=rs4.getString("Empname");

                String s4=rs4.getString("Empsalary");

                  System.out.println("\n"+r4+"\t"+n4+"\t"+s4);

                con.close();

                st.close();

                rs.close();

            }

        }

        catch(Exception e)

        {

        }

    }

}
```

OUTPUT:



```
D:\>javac Employs.java

D:\>java Employs
Enter Empno
8
Enter Empname
Krish
Enter Empsalary
8000
Empno   Empname Empsalary

4       Vinu    10000

5       Niya    6000

6       Diya    10000

7       Sreya   13000

8       Krish   8000
Enter Empno to be deleted
8
Empno    Empname         Empsalary

4       Vinu    10000

5       Niya    6000

6       Diya    10000

7       Sreya   13000
Enter Empno to be updated
4
Empno   Empname          Empsalary

4       Vinu    11000
```

PGM NO:2

RECORD SCROLLING FUNCTIONS

DATE:29/06/2016

AIM:

JDBC program to connect to Student table. Implement the record scrolling functions – first(),

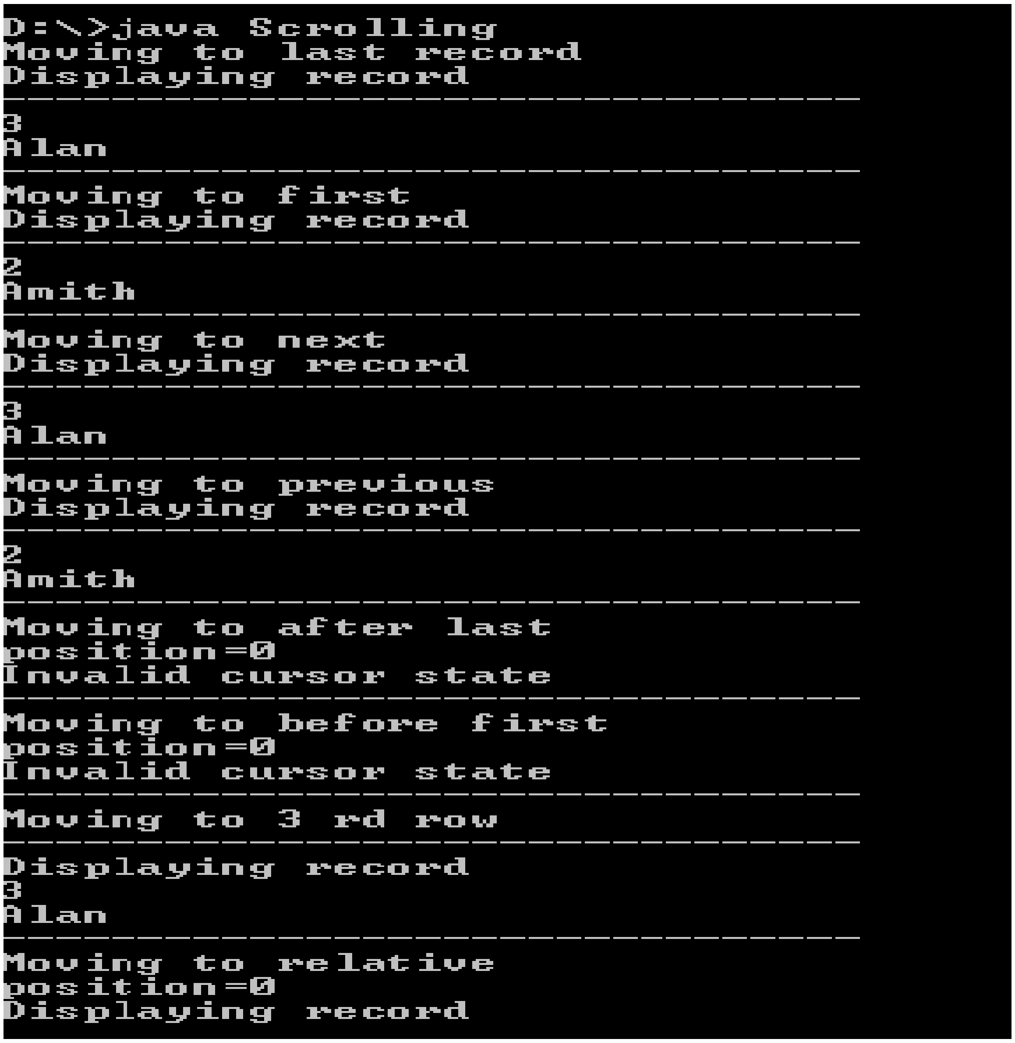
last(), next(), previous(), beforeFirst(), afterLast(), absolute() and relative().

PROGRAM:

```
import java.sql.*;

import java.io.*;

class Scrolling

{

        public static void main(String as[])throws ClassNotFoundException,IOException

        {

                try

                {

                        Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

                        Connection con=DriverManager.getConnection("jdbc:odbc:Data");

                        Statement st=con.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE,ResultSet.CONCUR_READ_ONLY);

                        ResultSet rs=st.executeQuery("select * from Table1");

                        System.out.println("Moving to last record");

                        rs.last();

                        System.out.println("Displaying record");
System.out.println("------------------------------");

System.out.println(rs.getString(1));

System.out.println(rs.getString(2));

System.out.println("------------------------------");

System.out.println("Moving to first");

rs.first();

System.out.println("Displaying record");

System.out.println("------------------------------");

System.out.println(rs.getString(1));

System.out.println(rs.getString(2));

System.out.println("------------------------------");

System.out.println("Moving to next");

rs.next();

System.out.println("Displaying record");

System.out.println("------------------------------");

System.out.println(rs.getString(1));

System.out.println(rs.getString(2));

System.out.println("------------------------------");

System.out.println("Moving to previous");

rs.previous();

System.out.println("Displaying record");

System.out.println("------------------------------");

System.out.println(rs.getString(1));

System.out.println(rs.getString(2));

System.out.println("------------------------------");
System.out.println("Moving to after last");
rs.afterLast();
int i=rs.getRow();
System.out.println("position="+i);
if(i==0)
System.out.println("Invalid cursor state");
System.out.println("------------------------------");
System.out.println("Moving to before first");
rs.beforeFirst();
int j=rs.getRow();
System.out.println("position="+j);
if(j==0)
System.out.println("Invalid cursor state");
System.out.println("------------------------------");
System.out.println("Moving to absolute");
System.out.println("------------------------------");
rs.absolute(-1);
System.out.println("Displaying record");
System.out.println(rs.getString(1));
System.out.println(rs.getString(2));
System.out.println("------------------------------");
System.out.println("Moving to relative");
rs.relative(-1);
int ii=rs.getRow();
System.out.println("position="+ii);
```

```
            System.out.println("Displaying record");
            System.out.println(rs.getString(1));
            System.out.println(rs.getString(2));
            con.close();
    }
    catch(SQLException e)
    {
        System.out.print(e);
    }
}
}
```

OUTPUT:



```
D:\>java Scrolling
Moving to last record
Displaying record
-------------------------------
3
Alan
-------------------------------
Moving to first
Displaying record
-------------------------------
2
Amith
-------------------------------
Moving to next
Displaying record
-------------------------------
3
Alan
-------------------------------
Moving to previous
Displaying record
-------------------------------
2
Amith
-------------------------------
Moving to after last
position=0
Invalid cursor state
-------------------------------
Moving to before first
position=0
Invalid cursor state
-------------------------------
Moving to 3 rd row
-------------------------------
Displaying record
3
Alan
-------------------------------
Moving to relative
position=0
Displaying record
```

PGM NO:3

BANK TRANSACTIONS

DATE:04/07/2016

AIM:

Create Bank table with fieldsacc_no, name, balance and insert records into it. Write a JDBC program to deposit and withdraw amounts from a particular account. Also implement transfer of amount from one account to another. Manage the transaction.

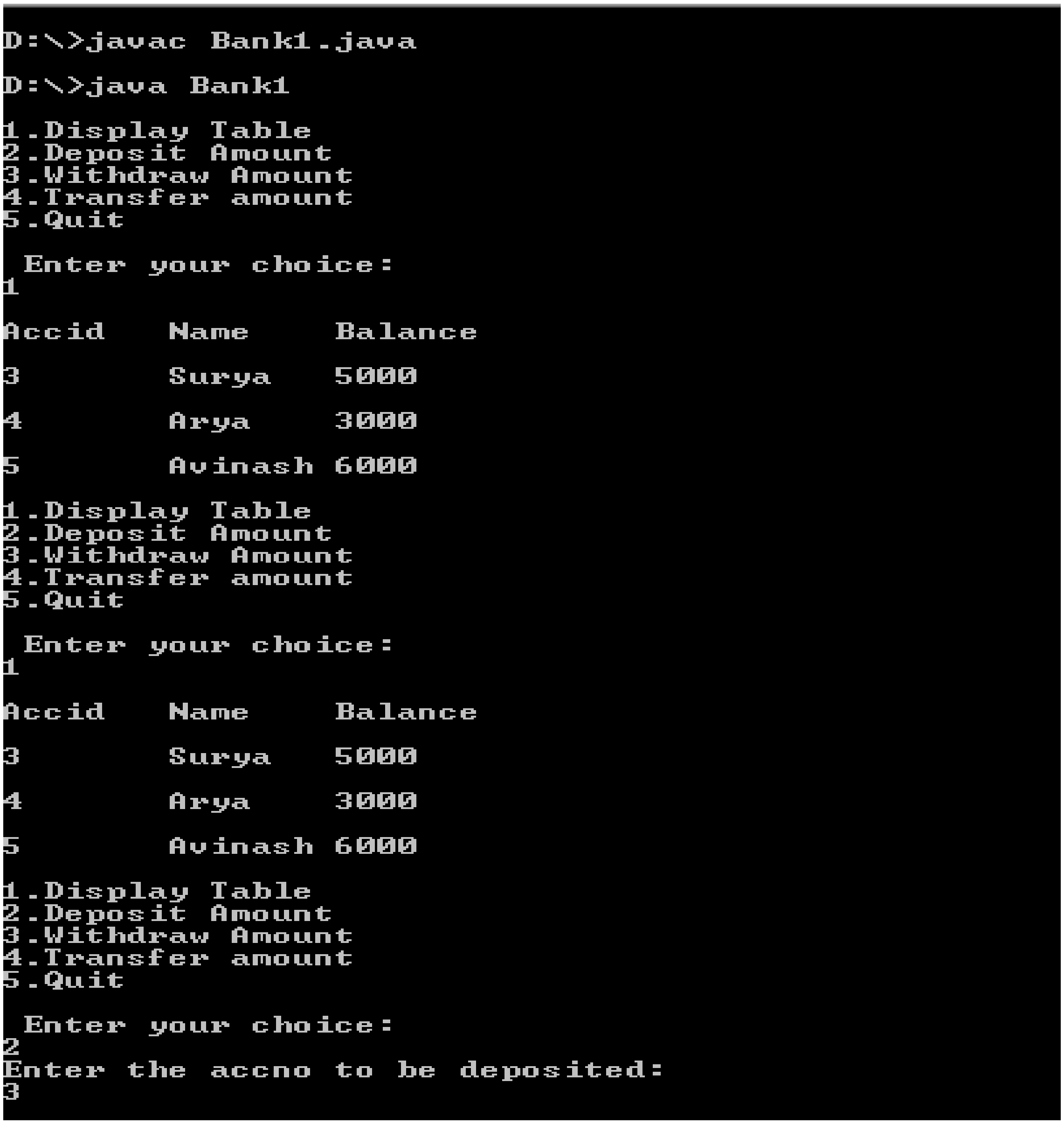
PROGRAM:

```
import java.sql.*;

import java.io.*;

class Bank

{

        public static void main(String args[])

        {

        try

        {

                int ch;

                Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

                Connection con=DriverManager.getConnection("jdbc:odbc:Bankdata");

                Statement st=con.createStatement();

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

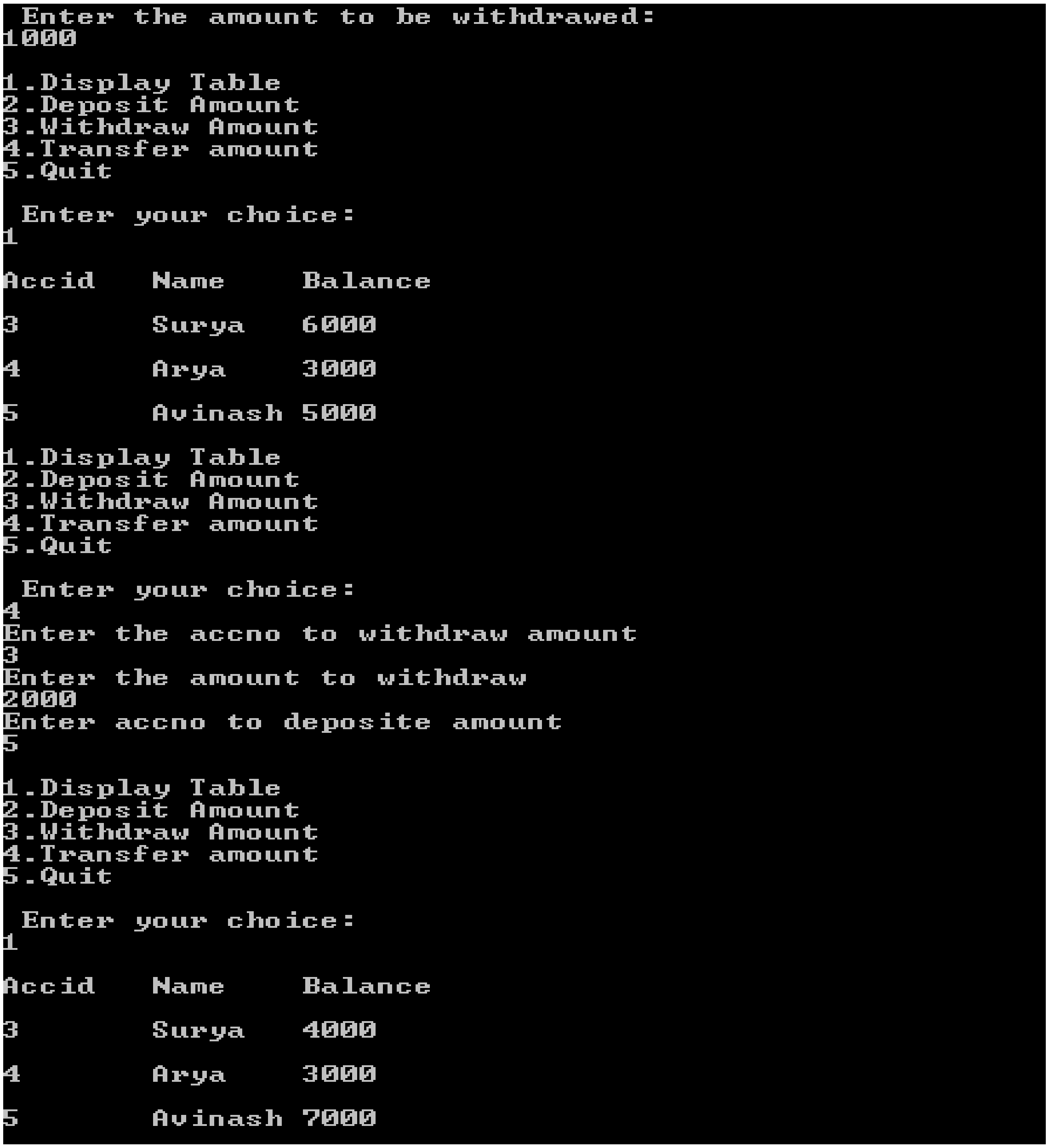
```
do
{
    System.out.println("\n1.Display Table\n2.Deposit Amount\n3.Withdraw Amount\n4.Transfer amount\n5.Quit");
    System.out.println("\n Enter your choice:");
    ch=Integer.parseInt(br.readLine());
    switch(ch)
    {
    case 1:
        ResultSet rs=st.executeQuery("select * from bank");
        System.out.println("\nAccno\tName\tBalance");
        while(rs.next())
        {
            String a=rs.getString("Accno");
            String n=rs.getString("Name");
            String b=rs.getString("Balance");
            System.out.println("\n"+a+"\t"+n+"\t"+b);
        }
        break;
    case 2:
        System.out.println("Enter the accno to be deposited:");
        int a1=Integer.parseInt(br.readLine());
        System.out.println("\n Enter the amout to be deposited:");
        int b1=Integer.parseInt(br.readLine());
        st.executeUpdate("update bank set Balance=Balance"+"+"+b1+" where Accno="+a1+";");
```

```
                break;

            case 3:

                System.out.println("Enter the accno to be withdrawed:");

                int a3=Integer.parseInt(br.readLine());

                System.out.println("\n Enter the amount to be withdrawed:");

                int b3=Integer.parseInt(br.readLine());

                ResultSet ba=st.executeQuery("select Balance from bank
                where Accno="+a3+";");

                int c=0;

                while(ba.next())

                {

                    c=Integer.parseInt(ba.getString("Balance"));

                }




                if(b3>c)

                {

                    System.out.println("Customer have insufficient
balance");

                    System.out.println("Customer balance="+c);

                }

                else

                {

                    st.executeUpdate("update bank set
                    Balance=Balance"+"-"+b3+" where Accno="+a3+";");
                }
                break;
            case  4:
                System.out.println("Enter the accno to withdraw amount");
                int a5=Integer.parseInt(br.readLine());
                System.out.println("Enter the amount to withdraw");
                int b5=Integer.parseInt(br.readLine());
                ResultSet ba1=st.executeQuery("select Balance from bank
                where Accno="+a5+";");
                int c1=0;
                while(ba1.next())
                {
                    c1=Integer.parseInt(ba1.getString("Balance"));
                }
                if(b5>c1)
                {
                    System.out.println("Customer have insufficient
balance");
                    System.out.println("Customer Balance="+c1);
                }
                else
                {
                    st.executeUpdate("update bank set
                    Balance=Balance"+"-"+b5+" where Accno="+a5+";");
                    System.out.println("Enter accno to deposite amount");
                    int a6=Integer.parseInt(br.readLine());
                    st.executeUpdate("update bank set
                    Balance=Balance"+"+"+b5+" where Accno="+a6+";");
                }
                break;
            case 5:
                break;
            default:
                    System.out.println("wrong choice");
            }
        }
        while(ch<=4);
        con.close();
    }
    catch(Exception e)
    {
    }
    }
}
```

OUTPUT:



```
D:\>javac Bank1.java

D:\>java Bank1

1.Display Table
2.Deposit Amount
3.Withdraw Amount
4.Transfer amount
5.Quit

 Enter your choice:
1

Accid   Name    Balance

3       Surya   5000

4       Arya    3000

5       Avinash 6000

1.Display Table
2.Deposit Amount
3.Withdraw Amount
4.Transfer amount
5.Quit

 Enter your choice:
1

Accid   Name    Balance

3       Surya   5000

4       Arya    3000

5       Avinash 6000

1.Display Table
2.Deposit Amount
3.Withdraw Amount
4.Transfer amount
5.Quit

 Enter your choice:
2
Enter the accno to be deposited:
3
```



```
 Enter the amount to be withdrawed:
1000

1.Display Table
2.Deposit Amount
3.Withdraw Amount
4.Transfer amount
5.Quit

 Enter your choice:
1

Accid    Name     Balance

3        Surya    6000

4        Arya     3000

5        Avinash 5000

1.Display Table
2.Deposit Amount
3.Withdraw Amount
4.Transfer amount
5.Quit

 Enter your choice:
4
Enter the accno to withdraw amount
3
Enter the amount to withdraw
2000
Enter accno to deposite amount
5

1.Display Table
2.Deposit Amount
3.Withdraw Amount
4.Transfer amount
5.Quit

 Enter your choice:
1

Accid    Name     Balance

3        Surya    4000

4        Arya     3000

5        Avinash 7000
```

PGM NO:4

DATABASE METADATA

DATE:14/07/2016

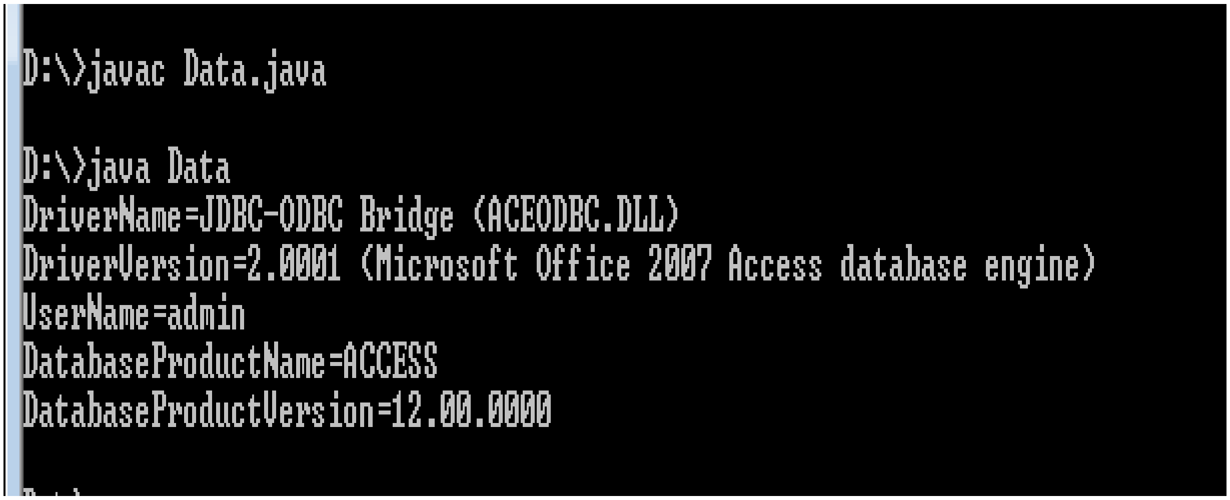
AIM: JDBC program to display database metadata.

PROGRAM:

```
import java.sql.*;

public class Data

{

        public static void main(String args[])

        {

                try

                {

                        Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

                        Connection
con=DriverManager.getConnection("jdbc:odbc:Database");

                        DatabaseMetaData dbmd=con.getMetaData();

                        System.out.println("DriverName="+dbmd.getDriverName());

                        System.out.println("DriverVersion="+dbmd.getDriverVersion());

                        System.out.println("UserName="+dbmd.getUserName());

System.out.println("DatabaseProductName="+dbmd.getDatabaseProductName());
```

```
        System.out.println("DatabaseProductVersion="+dbmd.getDatabaseProductVersion());
                    con.close();
            }
            catch(Exception e)
            {
                System.out.println("Exception"+e);
            }}  }
```

OUTPUT:



```
D:\>javac Data.java

D:\>java Data
DriverName=JDBC-ODBC Bridge (ACEODBC.DLL)
DriverVersion=2.0001 (Microsoft Office 2007 Access database engine)
UserName=admin
DatabaseProductName=ACCESS
DatabaseProductVersion=12.00.0000
```

PGM NO: 5

RESULTSET METADATA

DATE:28/07/2016

AIM: JDBC program to display database metadata

PROGRAM:

```
import java.sql.*;

public class Result

{       public static void main(String args[])

            {   try

                        {       Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

                                Connection con=DriverManager.getConnection("jdbc:odbc:Resultset");

                                Statement st=con.createStatement();

                                ResultSet rs=st.executeQuery("select * from Result");

                                ResultSetMetaData rsmd=rs.getMetaData();

                                System.out.println("column count="+rsmd.getColumnCount());

                                System.out.println("ColumnName="+rsmd.getColumnName(1));
```

PGMNO:6

DATE: 15/08/2016 **COMPLEX NUMBER OPERATIONS**

```
                        System.out.println("Column type
                        name="+rsmd.getColumnTypeName(2));

                        con.close();

                        st.close();

                        rs.close();

                }

                catch(Exception e)

                {       System.out.println("Exception"+e);

                }

        }  }
```

AIM:

RMI program for complex number operation

PROGRAM:

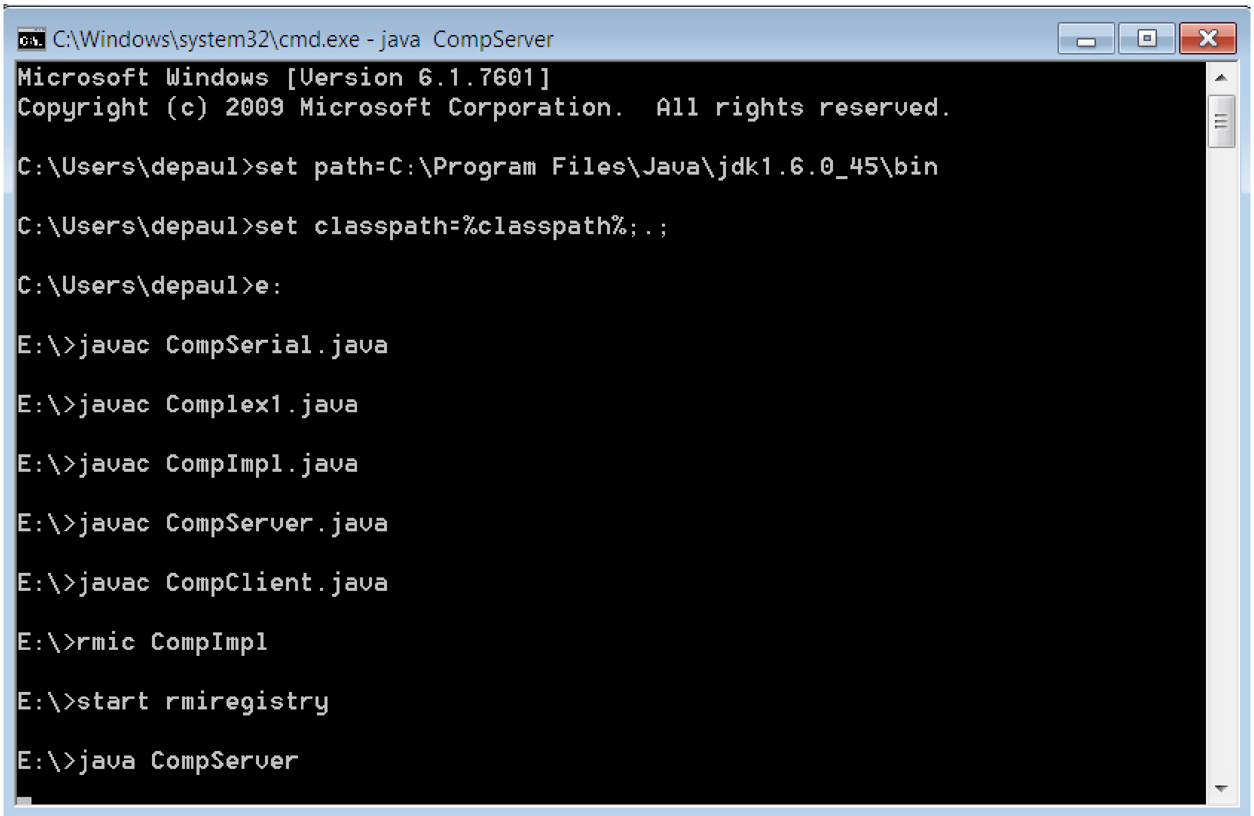
```
//CompSerial.java

import java.rmi.*;

import java.rmi.server.*;

import java.io.*;

public class CompSerial implements Serializable

{

        int real,imag;

        public CompSerial(int a,int b)

        {

                real=a;
        imag=b;
    }
}


//Complex1.java
import java.rmi.*;
import java.rmi.server.*;
public interface Complex1 extends Remote
{
    public CompSerial add(CompSerial c1,CompSerial c2)throws Exception;
    public CompSerial subtract(CompSerial c1,CompSerial c2)throws Exception;
    public CompSerial multiply(CompSerial c1,CompSerial c2)throws Exception;
    public CompSerial division(CompSerial c1,CompSerial c2)throws Exception;
}

//CompImpl.java
import java.rmi.*;
import java.rmi.server.*;
import java.io.*;
public class CompImpl extends UnicastRemoteObject implements Complex1
{
    CompSerial cs;
    public CompImpl()throws RemoteException,IOException
    {
        cs=new CompSerial(0,0);
```

```
}
public CompSerial add(CompSerial c1,CompSerial c2)throws Exception
{
        cs.real=c1.real+c2.real;
        cs.imag=c1.imag+c2.imag;
        return cs;
}
public CompSerial subtract(CompSerial c1,CompSerial c2)throws Exception
{
        cs.real=c1.real-c2.real;
        cs.imag=c1.imag-c2.imag;
        return cs;
}
public CompSerial multiply(CompSerial c1,CompSerial c2)throws Exception
{
        cs.real = c1.real * c2.real - c1.imag * c2.imag;
        cs.imag = c1.real * c2.imag + c1.imag * c2.real;
        return cs;

}
public CompSerial division(CompSerial c1,CompSerial c2)throws Exception
{
        cs.real=(c1.real*c2.real+c1.imag*c2.imag)/(c2.real*c2.real+c2.imag*c2.imag);
        cs.imag=(c1.imag*c2.real-c1.real*c2.imag)/(c2.real*c2.real+c2.imag*c2.imag);
        return cs;
```

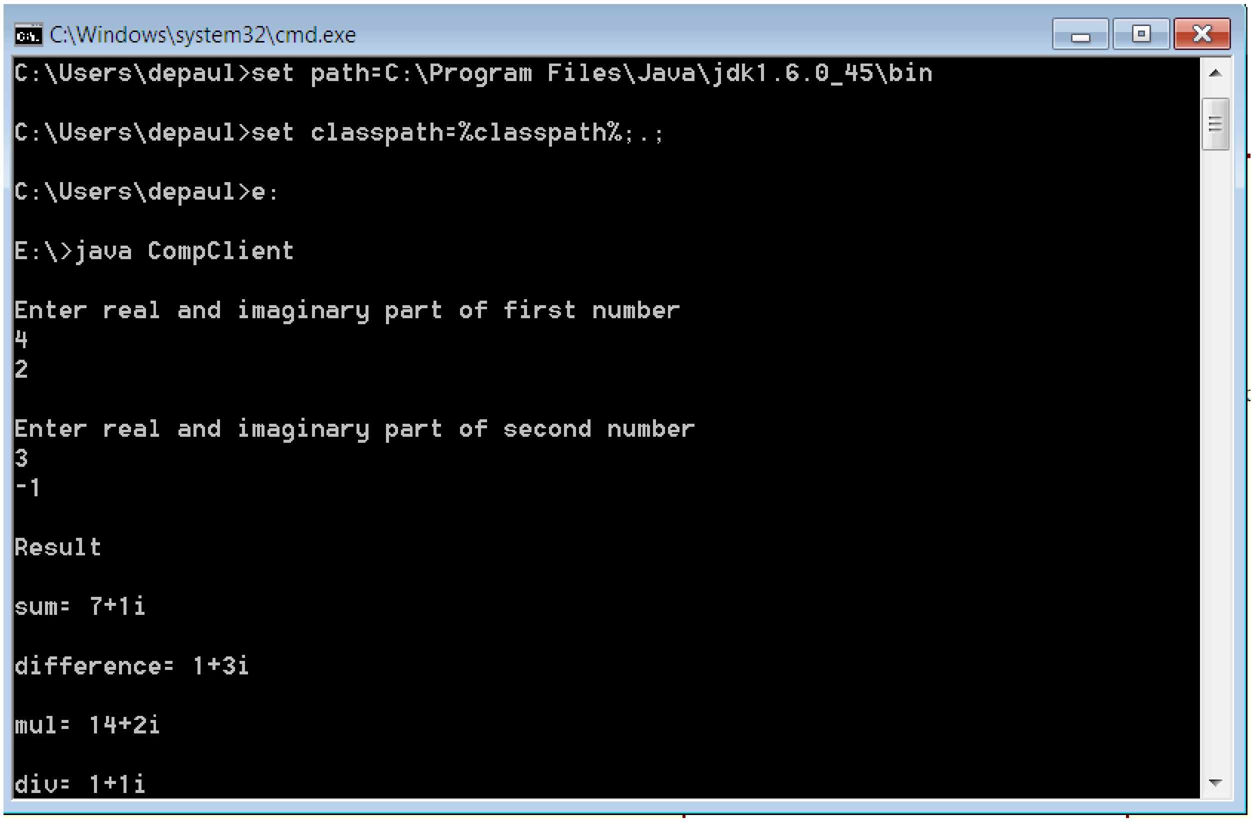
```
    }
}

  //CompServer.java

import java.rmi.*;
import java.rmi.server.*;
public class CompServer
{
    public static void main(String args[])
    {
        try
        {
            CompImpl cs=new CompImpl();
            Naming.rebind("rmi://127.0.0.1:1099/CompServer",cs);
        }
        catch(Exception e)
        {
            System.out.println(e);
        }
    }
}

//CompClient.java

import java.rmi.*;
import java.io.*;

public class CompClient

{

        public static void main(String args[])

        {

                try

                {

                        Complex1 com=(Complex1)Naming.lookup("//127.0.0.01:1099/CompServer");

                        BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

                        System.out.println("\nEnter real and imaginary part of first number");

                        int r1=Integer.parseInt(br.readLine());

                        int i1=Integer.parseInt(br.readLine());

                        System.out.println("\nEnter real and imaginary part of second number");

                        int r2=Integer.parseInt(br.readLine());

                        int i2=Integer.parseInt(br.readLine());

                        CompSerial cs1=new CompSerial(0,0);

                        CompSerial cs2=new CompSerial(r1,i1);

                        CompSerial cs3=new CompSerial(r2,i2);

                        System.out.println("\nResult");

                        cs1=com.add(cs2,cs3);

                        System.out.println("\nsum= "+cs1.real+"+"+cs1.imag+"i");

                        cs1=com.subtract(cs2,cs3);

                        System.out.println("\ndifference= "+cs1.real+"+"+cs1.imag+"i");

                        cs1=com.multiply(cs2,cs3);

                        System.out.println("\nmul= "+cs1.real+"+"+cs1.imag+"i");
```

```
            cs1=com.division(cs2,cs3);
            System.out.println("\ndiv= "+cs1.real+"+"+cs1.imag+"i");
        }
        catch(Exception e)
        {
            System.out.println("\nException= "+e);
        }
    }
}
```

OUTPUT:



```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\depaul>set path=C:\Program Files\Java\jdk1.6.0_45\bin

C:\Users\depaul>set classpath=%classpath%;.;

C:\Users\depaul>e:

E:\>javac CompSerial.java

E:\>javac Complex1.java

E:\>javac CompImpl.java

E:\>javac CompServer.java

E:\>javac CompClient.java

E:\>rmic CompImpl

E:\>start rmiregistry

E:\>java CompServer
```



```
C:\Users\depaul>set path=C:\Program Files\Java\jdk1.6.0_45\bin

C:\Users\depaul>set classpath=%classpath%;.;

C:\Users\depaul>e:

E:\>java CompClient

Enter real and imaginary part of first number
4
2

Enter real and imaginary part of second number
3
-1

Result

sum= 7+1i

difference= 1+3i

mul= 14+2i

div= 1+1i
```

PGMNO:7

DATE: 18/08/2016

BANK OPERATIONS

AIM:

RMI program for bank operation

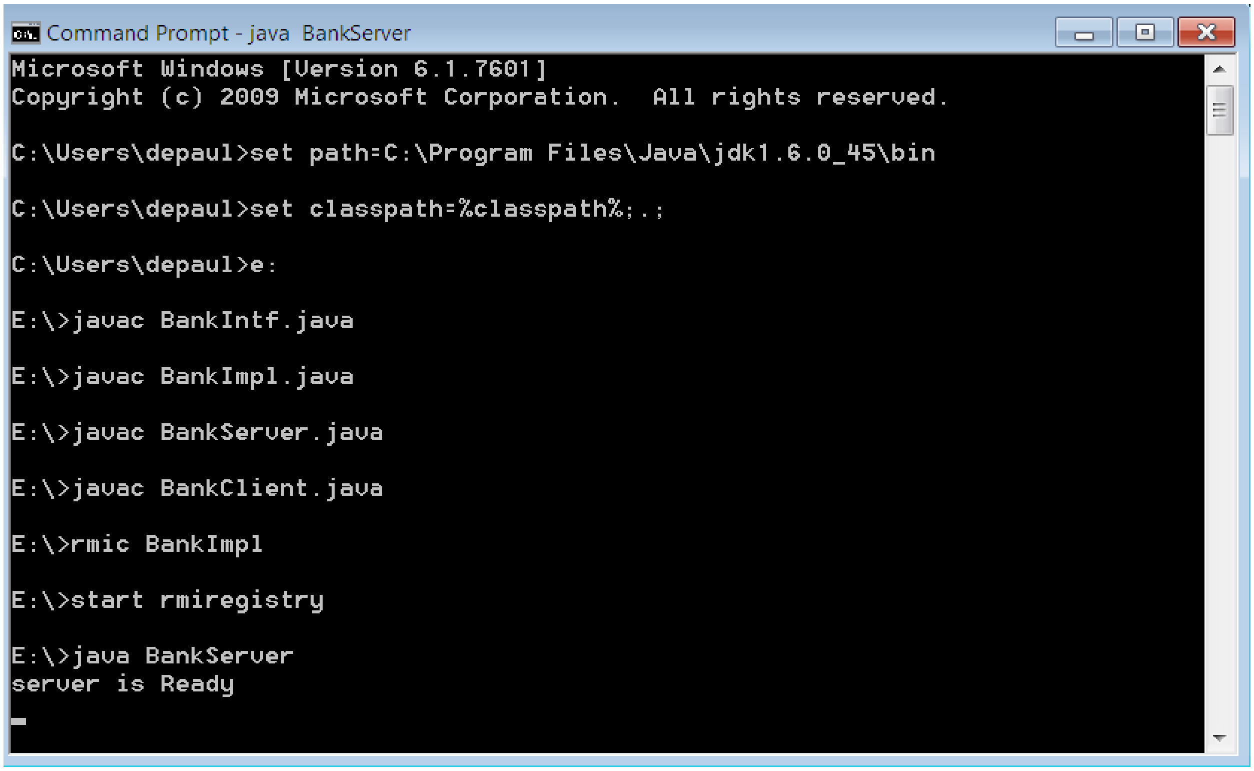
PROGRAM:

```
//BankIntf.java
import java.rmi.*;
public interface BankIntf extends Remote
{
    int withdraw(int a,int amt)throws RemoteException;
    int deposit(int a,int amt)throws RemoteException;
    int balance(int amt)throws RemoteException;
}


//BankImpl.java

import java.rmi.*;
import java.rmi.server.*;
public class BankImpl extends UnicastRemoteObject implements BankIntf
{
    public BankImpl()throws RemoteException
    {}
    public int withdraw(int a,int amt)throws RemoteException
```

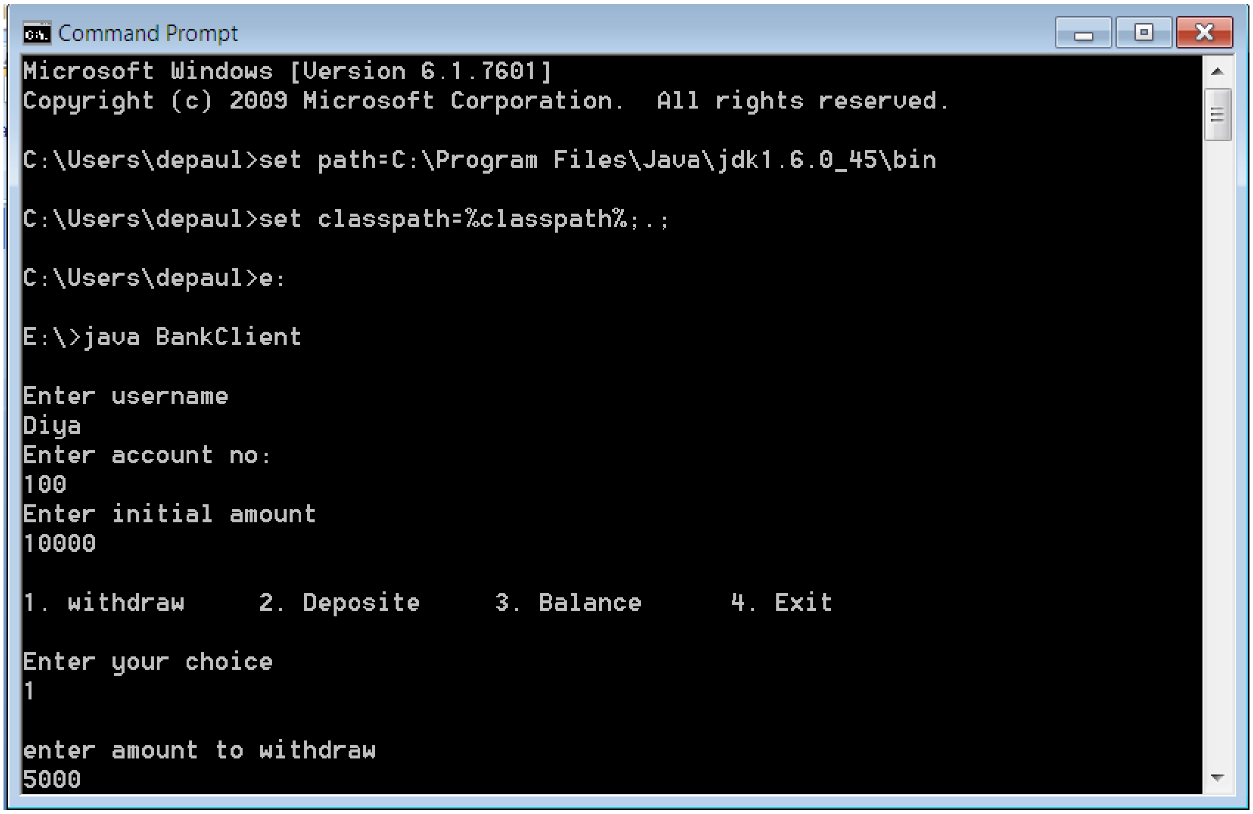
```
        {
                amt=amt-a;
                return (amt);
        }
        public int deposit(int a,int amt)throws RemoteException
        {
                amt=amt+a;
                return (amt);
        }
        public int balance(int amt)throws RemoteException
        {
                return (amt);
        }
}



//BankServer.java
import java.rmi.*;
public class BankServer
{
        public static void main(String args[])
        {
                try
                {
                        BankImpl bankimpl=new BankImpl();
                        Naming.rebind("BankServer",bankimpl);
                System.out.println("server is Ready");
            }
            catch(Exception e)
            {
                System.out.println(e);
            }
        }
    }

    //BankClient.java
    import java.rmi.*;
    import java.io.*;
    public class BankClient
    {
        public static void main(String args[])
        {
            int ch;
            try
            {
                BankIntf bankintf=(BankIntf
     )Naming.lookup("//127.0.0.01:1099/BankServer");
                BufferedReader br=new BufferedReader(new
    InputStreamReader(System.in));
                System.out.println("\nEnter username");
                String s=br.readLine();
                System.out.println("Enter account no: ");
                int ac=Integer.parseInt(br.readLine());
        System.out.println("Enter initial amount");
        int amt=Integer.parseInt(br.readLine());
        do
        {
            System.out.println("\n1. withdraw\t2. Deposite\t3. Balance\t4. Exit");
            System.out.println("\nEnter your choice");
            ch=Integer.parseInt(br.readLine());
            switch(ch)
            {
                case 1:
                System.out.println("\nenter amount to withdraw");
                int wd=Integer.parseInt(br.readLine());
                System.out.println("\nusername: "+s);
                System.out.println("\nAccount no: "+ac);
                if(wd>amt)
                {
                    System.out.println("\nInsufficient amount");
                }
                else
                {
                    amt=bankintf.withdraw(wd,amt);
                    System.out.println("Balance: "+amt);
                }
                break;
                case 2:
                System.out.println("\nenter amount to deposite");
```

```
                    int dp=Integer.parseInt(br.readLine());

                    System.out.println("\nusername: "+s);

                    System.out.println("\nAccountNo: "+ac);

                    amt=bankintf.deposit(dp,amt);

                    System.out.println("Balance: "+amt);

                    break;

                    case 3:

                    System.out.println("\nusername: "+s);

                    System.out.println("\nAccountNo: "+ac);

                    amt=bankintf.balance(amt);

                    System.out.println("Balance: "+amt);

                    break;

                }

            }

            while(ch<4);

        }

        catch(Exception e)

        {

            System.out.println("exception: "+e);

        }

    }

}
```

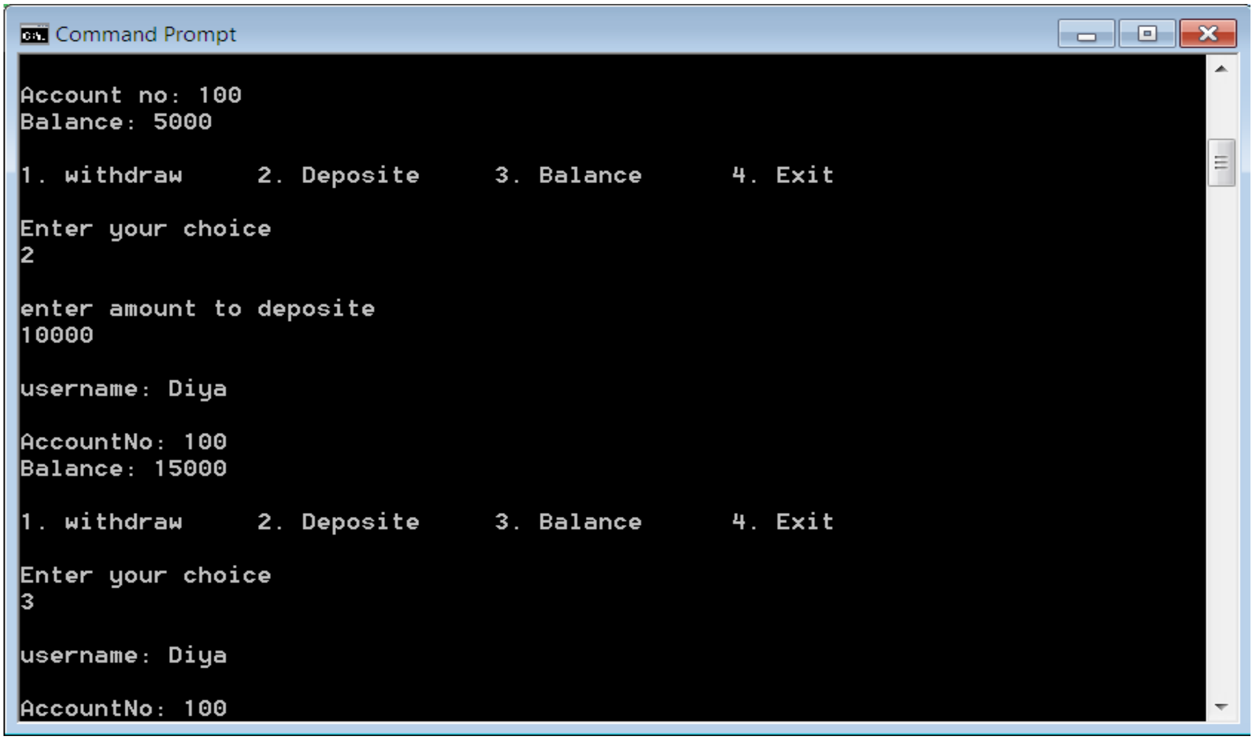
OUTPUT:



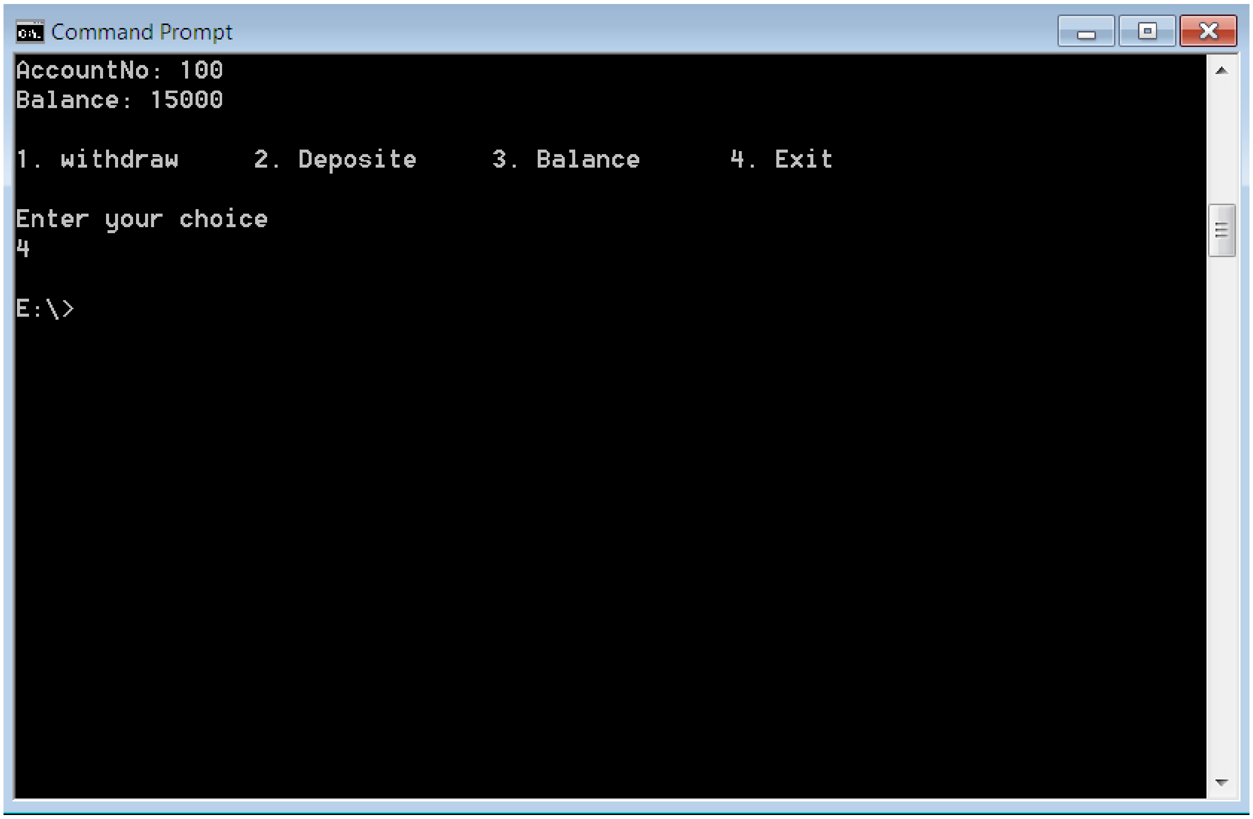
```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\depaul>set path=C:\Program Files\Java\jdk1.6.0_45\bin

C:\Users\depaul>set classpath=%classpath%;.;

C:\Users\depaul>e:

E:\>javac BankIntf.java

E:\>javac BankImpl.java

E:\>javac BankServer.java

E:\>javac BankClient.java

E:\>rmic BankImpl

E:\>start rmiregistry

E:\>java BankServer
server is Ready
```



```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\depaul>set path=C:\Program Files\Java\jdk1.6.0_45\bin

C:\Users\depaul>set classpath=%classpath%;.;

C:\Users\depaul>e:

E:\>java BankClient

Enter username
Diya
Enter account no:
100
Enter initial amount
10000

1. withdraw     2. Deposite     3. Balance      4. Exit

Enter your choice
1

enter amount to withdraw
5000
```



```

Account no: 100
Balance: 5000

1. withdraw     2. Deposite     3. Balance      4. Exit

Enter your choice
2

enter amount to deposite
10000

username: Diya

AccountNo: 100
Balance: 15000

1. withdraw     2. Deposite     3. Balance      4. Exit

Enter your choice
3

username: Diya

AccountNo: 100
```



```
AccountNo: 100
Balance: 15000

1. withdraw     2. Deposite     3. Balance      4. Exit

Enter your choice
4

E:\>
```

PGMNO: 8

DATE: 30/09/2016

MATRIX ADDITION AND SUBTRACTION

AIM:

RMI program for matrix addition and subtraction

PROGRAM:

```
//MatrixOp.java

import java.rmi.*;

import java.io.*;

public interface MatrixOp extends Remote
```

```
{
    public String add(int a[][],int b[][],int r,int c)throws RemoteException,IOException;
    public String sub(int a[][],int b[][],int r,int c)throws RemoteException,IOException;
}


//MatrixImpl.java
import java.rmi.*;
import java.io.*;
import java.rmi.server.*;
public class MatrixImpl extends UnicastRemoteObject implements MatrixOp
{
    public MatrixImpl()throws RemoteException
    {}
    public String add(int a[][],int b[][],int r,int c)throws RemoteException,IOException
    {
        int res[][]=new int[r][c];
        int i,j;
        String str;
        str="Addition of two matrices are"+"\n";
        for(i=0;i<r;i++)
        {
            for(j=0;j<c;j++)
            {
                res[i][j]=a[i][j]+b[i][j];
                //System.out.println("\n");
```

```
                                        str=str+"\t"+String.valueOf(res[i][j]);
                                        //System.out.println("\n");
                                }
                                str=str+"\n";
                        }
                        return str;
                }
                public String sub(int a[][],int b[][],int r,int c)throws RemoteException,IOException
                {
                        int res[][]=new int[r][c];
                        int i,j;
                        String str;
                        str="Subtraction of two matrices are"+"\n";
                        for(i=0;i<r;i++)
                        {
                                for(j=0;j<c;j++)
                                {
                                        res[i][j]=a[i][j]-b[i][j];
                                        //System.out.println("\n");
                                        str=str+"\t"+String.valueOf(res[i][j]);
                                }
                                str=str+"\n";
                        }
                        return str;
                }
        }
```

```
//MatrixServer.java
import java.rmi.*;
import java.rmi.server.*;
public class MatrixServer
{
        public static void main(String args[])
        {
                try
                {
                        MatrixImpl ob=new MatrixImpl();
                        Naming.rebind("MatrixServer",ob);
                        System.out.println("server is Ready");
                }
                catch(Exception e)
                {
                        System.out.println(e);
                }
        }
}
```

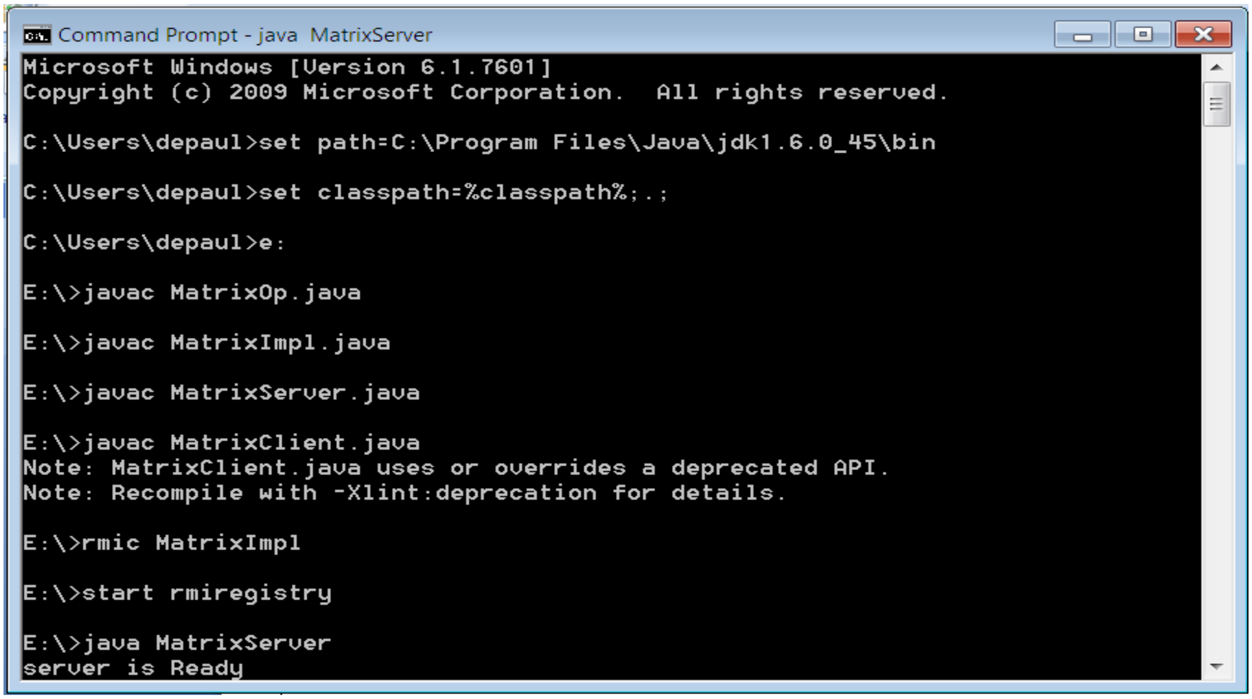
```
//MatrixClient.java
import java.rmi.*;
import java.io.*;
public class MatrixClient
{
        public static void main(String args[])
```

```
    {
        try
        {
            int i,j,row,col;
            String str=new String("");
            DataInputStream d=new DataInputStream(System.in);
            MatrixOp
ob=(MatrixOp)Naming.lookup("//127.0.0.01:1099/MatrixServer");
            System.out.println("1- Addition\t2- Subtraction");
            System.out.println("Enter choice");
            int ch=Integer.parseInt(d.readLine());
            switch(ch)
            {
                case 1:
                System.out.println("enter rows & columns");
                row=Integer.parseInt(d.readLine());
                col=Integer.parseInt(d.readLine());
                int a[][]=new int[row][col];
                int b[][]=new int[row][col];
                System.out.println("enter values of matrix A");
                for(i=0;i<row;i++)
                {
                    for(j=0;j<col;j++)
                    {
                        a[i][j]=Integer.parseInt(d.readLine());
                    }
                }
```

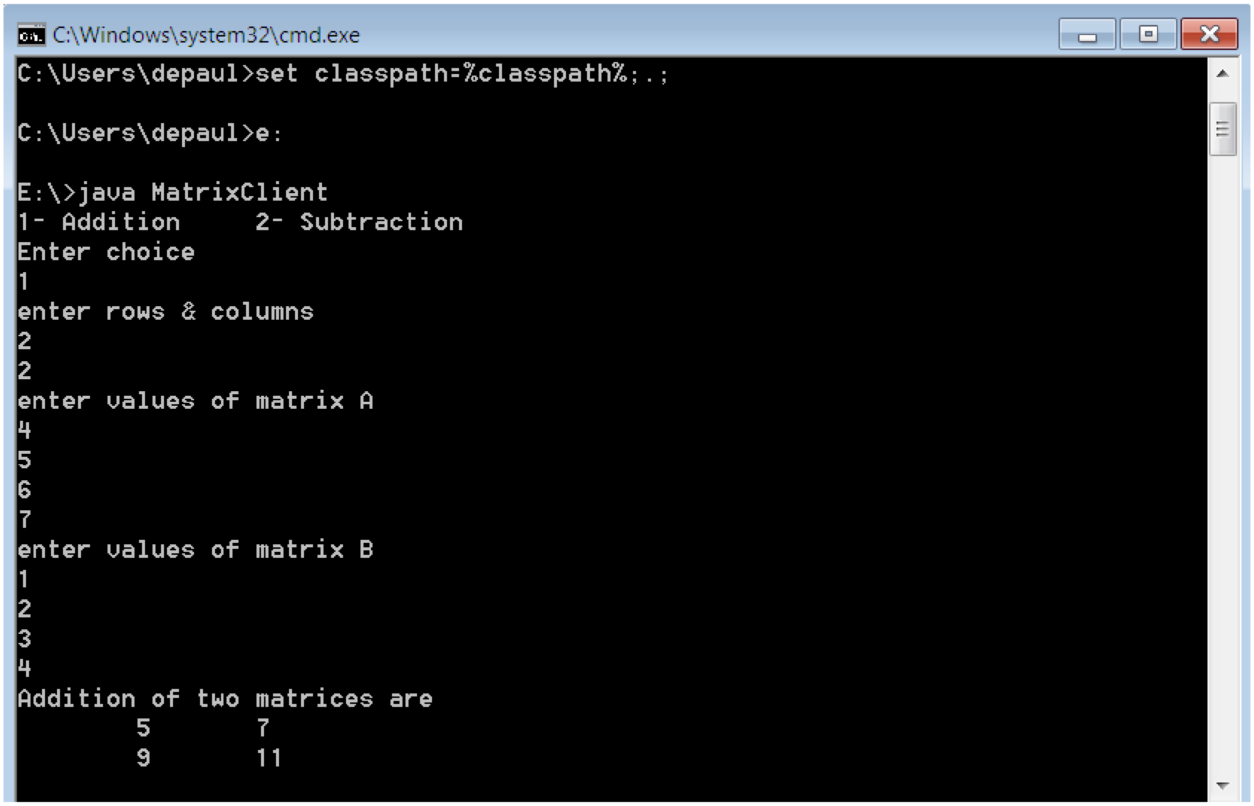
```
System.out.println("enter values of matrix B");
for(i=0;i<row;i++)
{
        for(j=0;j<col;j++)
        {
                b[i][j]=Integer.parseInt(d.readLine());
        }
}
str=(String)(ob.add(a,b,row,col));
break;
case 2:
System.out.println("enter rows & columns");
row=Integer.parseInt(d.readLine());
col=Integer.parseInt(d.readLine());
int a1[][]=new int[row][col];
int b1[][]=new int[row][col];
System.out.println("enter values of matrix A");
for(i=0;i<row;i++)
{
        for(j=0;j<col;j++)
        {
                a1[i][j]=Integer.parseInt(d.readLine());
        }
}
System.out.println("enter values of matrix B");
for(i=0;i<row;i++)
```

```
                {
                    for(j=0;j<col;j++)
                    {
                        b1[i][j]=Integer.parseInt(d.readLine());
                    }
                }
                str=(String)(ob.sub(a1,b1,row,col));
                break;
            }
                System.out.println(str);
        }
        catch(Exception e)
        {
            System.out.println(e);
        }
    }
```

OUTPUT:



```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\depaul>set path=C:\Program Files\Java\jdk1.6.0_45\bin

C:\Users\depaul>set classpath=%classpath%;.;

C:\Users\depaul>e:

E:\>javac MatrixOp.java

E:\>javac MatrixImpl.java

E:\>javac MatrixServer.java

E:\>javac MatrixClient.java
Note: MatrixClient.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.

E:\>rmic MatrixImpl

E:\>start rmiregistry

E:\>java MatrixServer
server is Ready
```



```
C:\Users\depaul>set classpath=%classpath%;.;

C:\Users\depaul>e:

E:\>java MatrixClient
1- Addition     2- Subtraction
Enter choice
1
enter rows & columns
2
2
enter values of matrix A
4
5
6
7
enter values of matrix B
1
2
3
4
Addition of two matrices are
        5       7
        9       11
```

PGM NO: 9

DATE: 02/09/2016

REQUEST INFORMATION

AIM

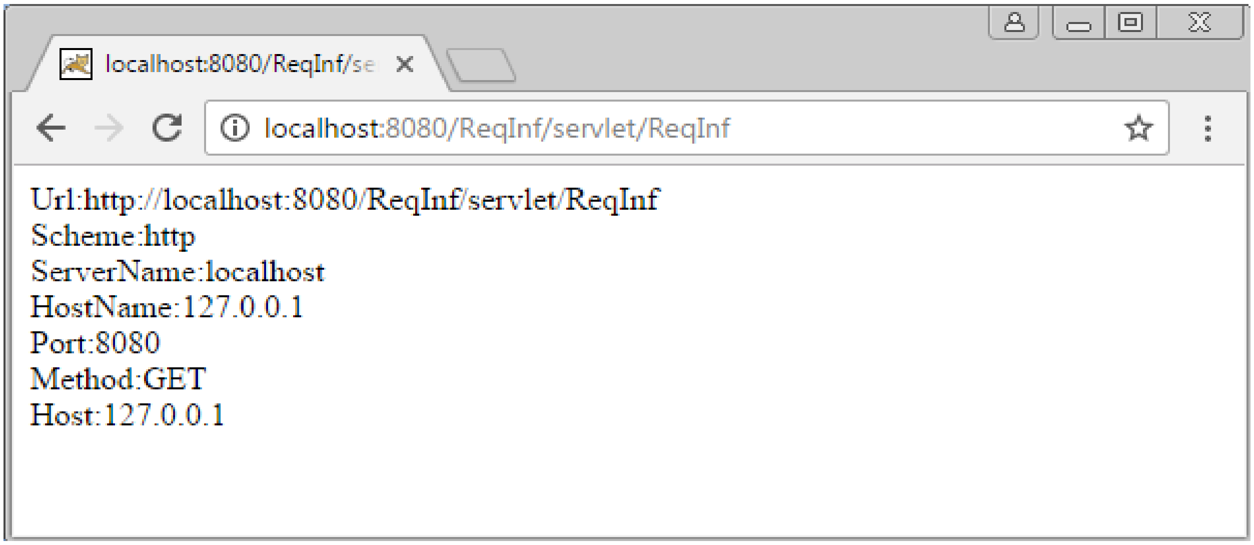
Servlet program that displays request information such as protocol,remote host name,server name,server port,header names,specific headers,authentication type etc.

PROGRAM

```
import javax.servlet.*;

import javax.servlet.http.*;

import java.io.*;

import java.util.*;

public class ReqInf extends HttpServlet
```

```
{
    protected void doGet(HttpServletRequest request,HttpServletResponse response)throws ServletException,IOException
    {
        doPost(request,response);
    }
        protected void doPost(HttpServletRequest request,HttpServletResponse response)throws ServletException,IOException
        {
            String url=request.getRequestURL().toString();
            String clientHost=request.getRemoteHost();
            String scheme=request.getScheme();
            String serverName=request.getServerName();
            String hostName=request.getRemoteHost();
            int portNumber=request.getServerPort();
            String meth=request.getMethod();
            response.setContentType("text/html");
            PrintWriter pw=response.getWriter();
            pw.print("Url:"+url+"<br/>");
            pw.print("Scheme:"+scheme+"<br/>");
            pw.print("ServerName:"+serverName+"<br/>");
            pw.print("HostName:"+hostName+"<br/>");
            pw.print("Port:"+portNumber+"<br/>");
            pw.print("Method:"+meth+"<br/>");
            pw.print("Host:"+clientHost+"<br/>");
        }
}
```

OUTPUT:



```
Url:http://localhost:8080/ReqInf/servlet/ReqInf
Scheme:http
ServerName:localhost
HostName:127.0.0.1
Port:8080
Method:GET
Host:127.0.0.1
```

PGM NO: 10
DATE: 07/09/2016

ATM SERVLET

AIM

ATM servlet to handle bank operations

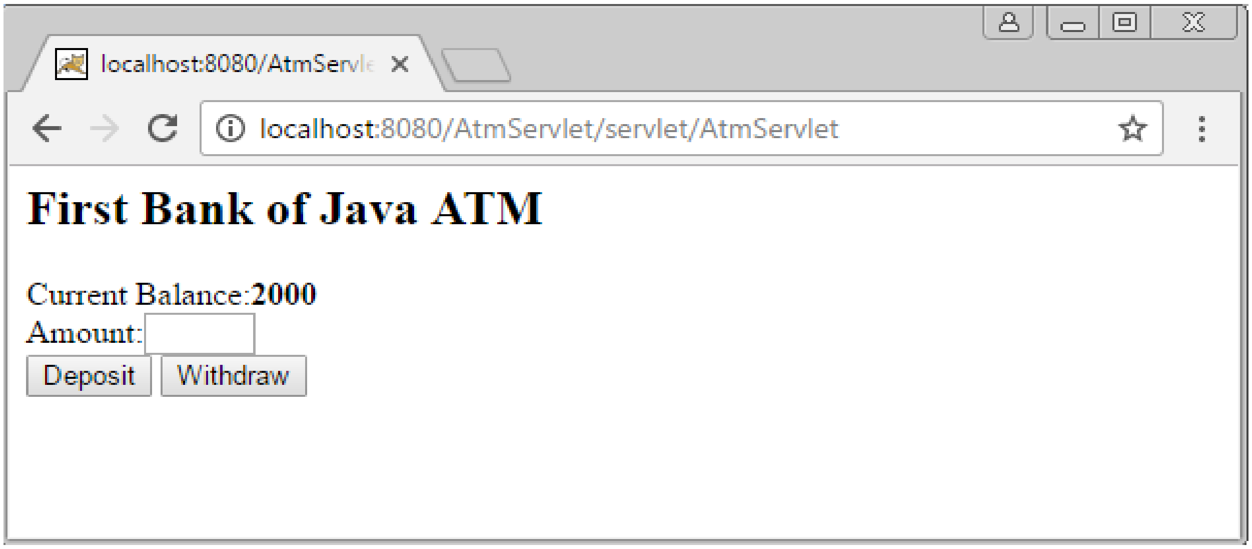
PROGRAM

```
import javax.servlet.*;

import javax.servlet.http.*;

import java.io.*;

import java.util.*;

public class AtmServlet extends HttpServlet
```

```
{
    Account act;
    public void init(ServletConfig conf)throws ServletException
    {
        super.init();
        act=new Account();
        act.balance=0;
    }
    public void doGet(HttpServletRequest req,HttpServletResponse resp)throws
ServletException,IOException
    {
        resp.setContentType("text/html");
        PrintWriter out=resp.getWriter();
        out.println("<HTML><BODY>");
        out.println("<H2>First Bank of Java ATM</H2>");
        out.println("Current Balance:<B>"+act.balance+"</B><BR>");
        out.println("<FORM METHOD=POST>");
        out.println("Amount:<INPUT TYPE=TEXT NAME=AMOUNT SIZE=3><BR>");
        out.println("<INPUT TYPE=SUBMIT NAME=DEPOSIT VALUE=Deposit>");
        out.println("<INPUT TYPE=SUBMIT NAME=WITHDRAW VALUE=Withdraw>");
        out.println("</FORM>");
        out.println("</BODY></HTML>");
    }
    protected void doPost(HttpServletRequest req,HttpServletResponse resp)throws
ServletException,IOException
    {
        int amt=0;
        try
        {
                amt=Integer.parseInt(req.getParameter("AMOUNT"));
        }
        catch(NullPointerException e)
        {
        }
        catch(NumberFormatException e)
        {
        }
        synchronized(act)
        {
                if((req.getParameter("WITHDRAW")!=null)&&(amt<act.balance))
                act.balance=act.balance-amt;
                if((req.getParameter("DEPOSIT")!=null)&&(amt>0))
                act.balance=act.balance+amt;
        }
        doGet(req,resp);
}
public void destroy()
{
}
class Account
{
        public int balance;
}
```

```
}
```

OUTPUT



First Bank of Java ATM

Current Balance:**2000**
Amount:

PGM NO:11

SESSION TRACKING

DATE: 11/09/2016

AIM

Session handling servlet that displays total number of visits to that page.

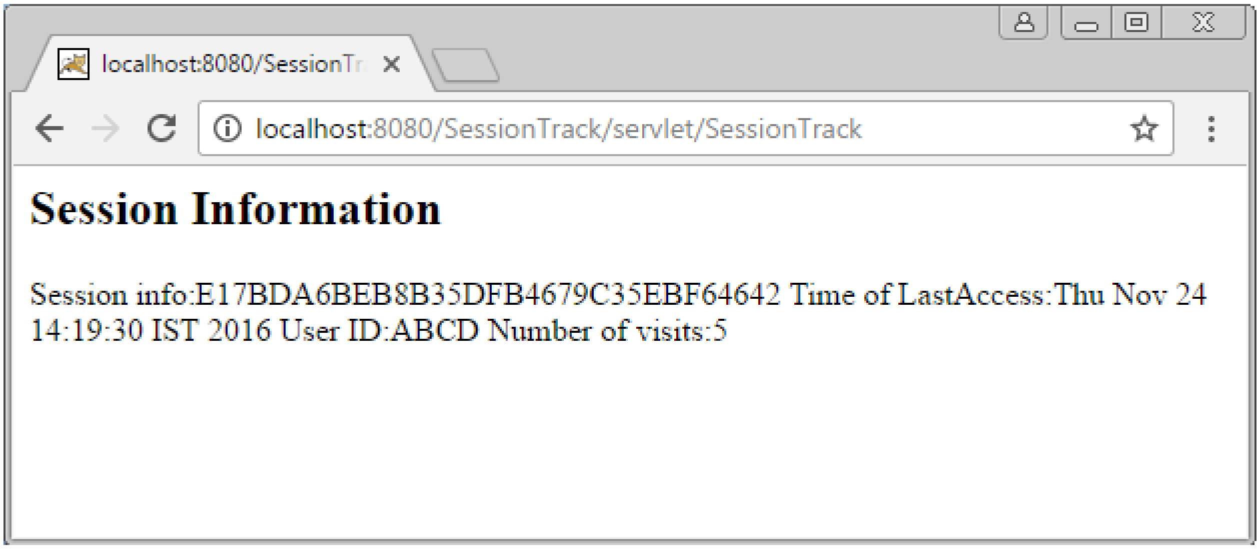
PROGRAM

```
import javax.servlet.*;

import javax.servlet.http.*;

import java.io.*;

import java.util.*;
```

```
public class SessionTrack extends HttpServlet
{
    public void doGet(HttpServletRequest request,HttpServletResponse response)throws ServletException,IOException
    {
        HttpSession session=request.getSession(true);
        Date creationTime=new Date(session.getCreationTime());
        Date lastAccessTime=new Date(session.getLastAccessedTime());
        String title="Welcome back to my website";
        Integer visitCount=new Integer(0);
        String visitCountKey=new String("visitCount");
        String userIDKey=new String("userID");
        String userID=new String("ABCD");
        if(session.isNew())
        {
            title="Welcome to my website";
            session.setAttribute(userIDKey,userID);
        }
        else
        {
            visitCount=(Integer)session.getAttribute(visitCountKey);
            visitCount=visitCount+1;
            userID=(String)session.getAttribute(userIDKey);
        }
        session.setAttribute(visitCountKey,visitCount);
        response.setContentType("text/html");
        PrintWriter out=response.getWriter();
```

```
                out.println("<h2>Session Information</h2>");
                out.println("Session info:"+session.getId());
                out.println("Time of LastAccess:"+lastAccessTime);
                out.println("User ID:"+userID);
                out.println("Number of visits:"+visitCount);
        }
}
```

OUTPUT



Session Information

Session info:E17BDA6BEB8B35DFB4679C35EBF64642 Time of LastAccess:Thu Nov 24 14:19:30 IST 2016 User ID:ABCD Number of visits:5

PGM NO:12

STUDENT INFORMATION

DATE: 15/09/2016

AIM

Create html form to read student details suchasroll, name, age, mark, sex, qualification etc. Write a servlet program that displays the same details.

PROGRAM

//HelloForm.html

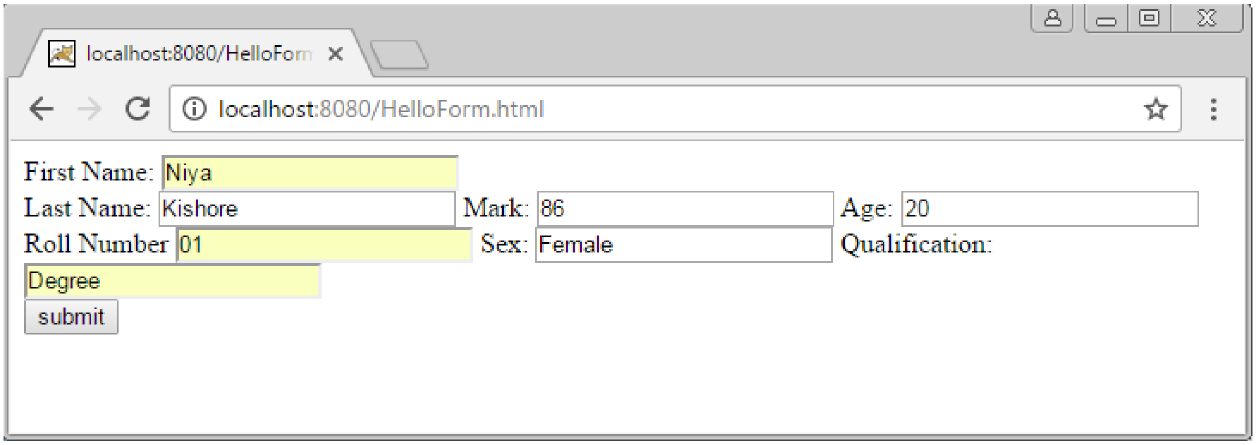
```
<html>

<body>

<form action=http://localhost:8080/HelloForm/servlet/HelloForm method="GET">
```

```
First Name: <input type="text" name="first_name"></br>

Last Name: <input type="text" name="last_name">

Mark: <input type="text" mark="mark">

Age: <input type="text" age="age">

Roll Number <input type="text" roll_number="roll_number">

Sex: <input type="text" sex="sex">

Qualification: <input type="text" qualification="qualification"></br>

<input type="submit" value="submit">

</form>

</body>

</html>
```

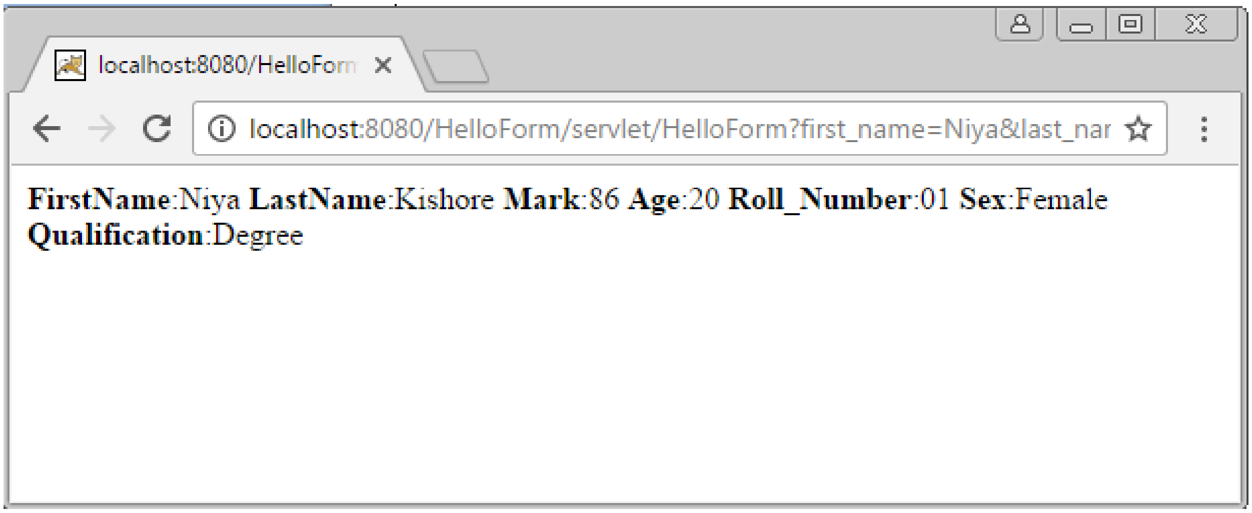
//HelloForm.java

```
import java.io.*;

import javax.servlet.*;

import javax.servlet.http.*;

public class HelloForm extends HttpServlet

{

        protected void doGet(HttpServletRequest request,HttpServletResponse response)throws ServletException,IOException

        {

                response.setContentType("text/html");
                PrintWriter out=response.getWriter();

                String title="using GET method to read fr4om data";

                out.println("<br>FirstName</br>:"+request.getParameter("first_name"));

                out.println("<br>LastName</br>:"+request.getParameter("last_name"));

                out.println("<br>Mark</br>:"+request.getParameter("mark"));

                out.println("<br>Age</br>:"+request.getParameter("age"));

                out.println("<br>Roll_Number</b>:"+request.getParameter("roll_number"));

                out.println("<br>Sex</br>:"+request.getParameter("sex"));

                out.println("<br>Qualification</br>:"+request.getParameter("qualification"));

        }

}
```

OUTPUT



First Name: Niya
Last Name: Kishore Mark: 86 Age: 20
Roll Number 01 Sex: Female Qualification:
Degree



localhost:8080/HelloForm
localhost:8080/HelloForm/servlet/HelloForm?first_name=Niya&last_nar

FirstName:Niya **LastName**:Kishore **Mark**:86 **Age**:20 **Roll_Number**:01 **Sex**:Female **Qualification**:Degree

PGM NO: 13 DATE: 20/09/2016	DISPLAY FILE CONTENT

AIM:

Servlet program that display the contents of a file specified by the user.

PROGRAM:

//TextFileRead.java

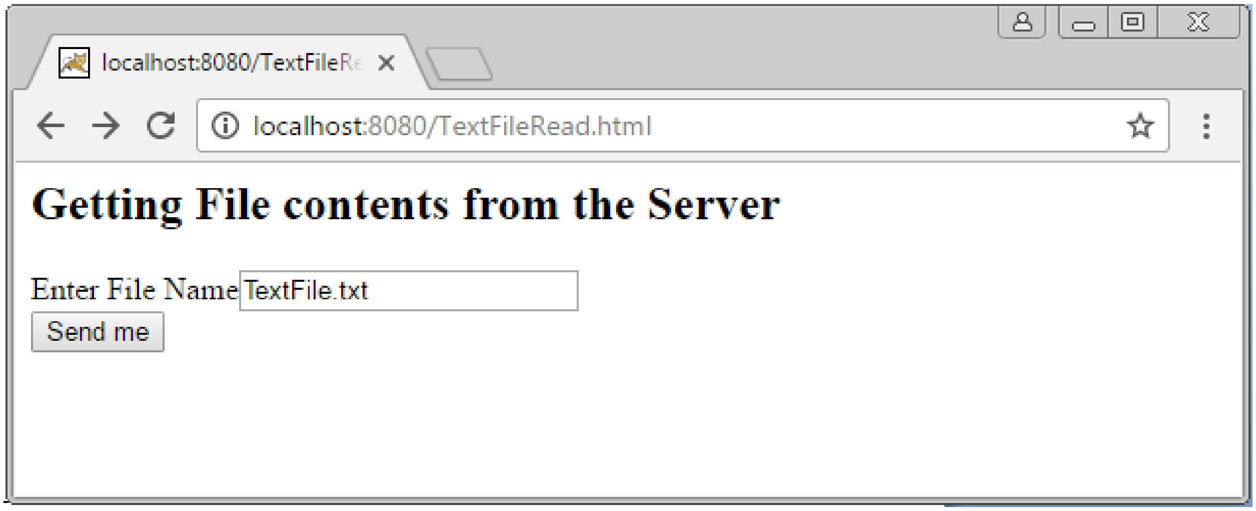
```
<BODY>

<H2>Getting File contents from the Server</H2>
```

```
<FORM METHOD="get" ACTION="http://localhost:8080/TextFileRead/servlet/TextFileRead">
Enter File Name<INPUT TYPE="text" NAME="filename"><BR>
<INPUT TYPE="submit"  VALUE="Send me">
</FORM>
</BODY>
```

TextFileRead.java

```
import javax.servlet.ServletException;
import javax.servlet.http.*;
import java.io.*;
public class TextFileRead extends HttpServlet
{
    public void service(HttpServletRequest req,HttpServletResponse res)throws ServletException,IOException
    {
        res.setContentType("text/html");
        PrintWriter pw=res.getWriter();
        String name=req.getParameter("filename");
        BufferedReader br=new BufferedReader(new FileReader("D:/"+name));
        String str;
        while((str=br.readLine())!=null)
        {
            pw.println(str+"<BR>");
        }
        br.close();
        pw.close();
        }
}
```

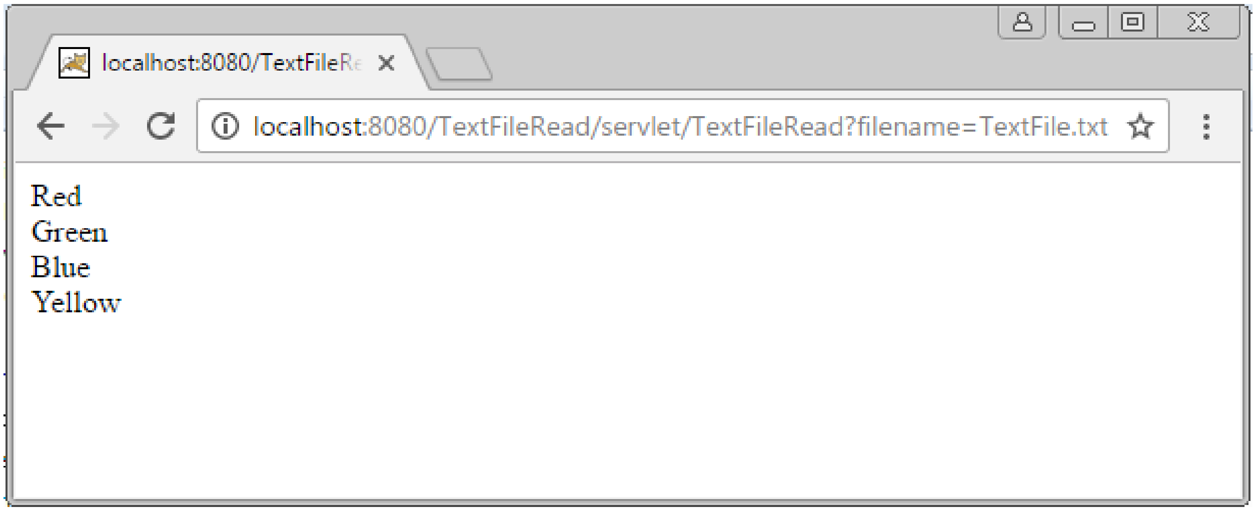
OUTPUT



localhost:8080/TextFileRead.html

Getting File contents from the Server

Enter File Name TextFile.txt


localhost:8080/TextFileRe
localhost:8080/TextFileRead/servlet/TextFileRead?filename=TextFile.txt

```
Red
Green
Blue
Yellow
```

PGMNO: 14
DATE: 07/10/2016

ARITHMETIC OPERATIONS

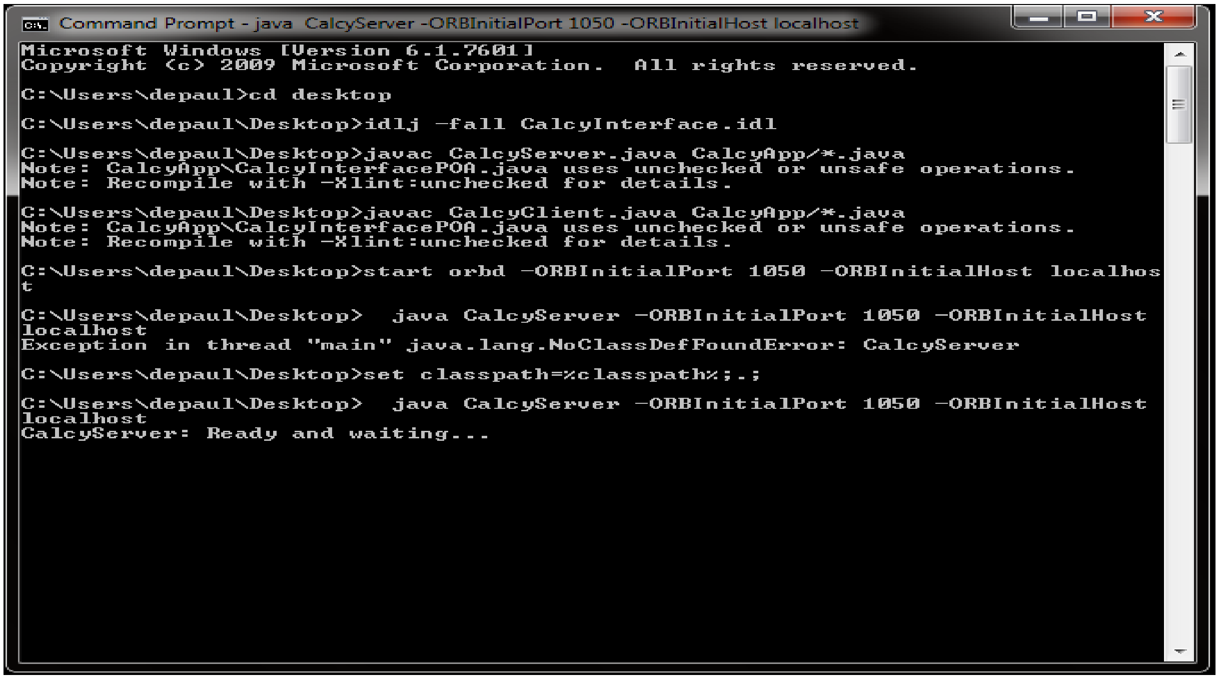
AIM: CORBA program for arithmetic operation.

PROGRAM

```
// CalcyInterface.idl

module CalcyApp

{

    interface CalcyInterface
    {
                double add(in double x, in double y);
                double subtract(in  double a, in  double b);
                double multiply(in  double x, in  double y);
                 double divide(in  double a, in  double b);
                oneway void shutdown();
    };
};

// CalcyServer.java

   import CalcyApp.*;
   import org.omg.CosNaming.*;
   import org.omg.CosNaming.NamingContextPackage.*;
   import org.omg.CORBA.*;
   import org.omg.PortableServer.*;
  import org.omg.PortableServer.POA;
  import java.util.Properties;
 class CalcyInterfaceImpl extends CalcyInterfacePOA
{
        private ORB orb;

        public void setORB(ORB orb_val)
        {
                orb = orb_val;
}
 public double add(double x, double y)
{
          double res= x+y;
          return (res);
 }
 public double subtract(double x, double y)
  {
          double res= x-y;
          return (res);
 }
public double multiply(double x, double y)
 {
          double res= x*y;
          return (res);
}
public double divide(double x, double y)
 {
          double res= x/y;
          return (res);
}
public void shutdown()
{
          orb.shutdown(false);
}
```

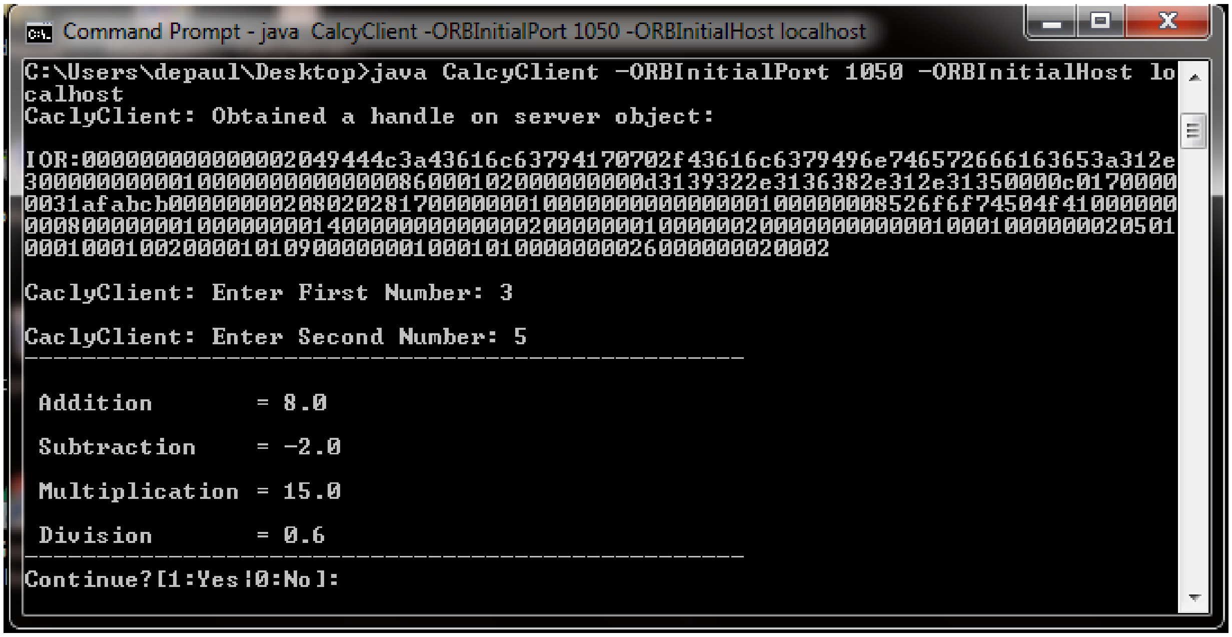
```
}
public class CalcyServer
{
  public static void main(String args[])
 {
try{
      ORB orb = ORB.init(args, null);
      POA rootpoa = POAHelper.narrow(orb.resolve_initial_references("RootPOA"));
    rootpoa.the_POAManager().activate();
       CalcyInterfaceImpl CalcyImpl = new CalcyInterfaceImpl();
       CalcyImpl.setORB(orb);


        org.omg.CORBA.Object ref = rootpoa.servant_to_reference(CalcyImpl);
      CalcyInterface href = CalcyInterfaceHelper.narrow(ref);
org.omg.CORBA.Object objRef =orb.resolve_initial_references("NameService");
NamingContextExt ncRef = NamingContextExtHelper.narrow(objRef);
String name = "CalcyOperations";
    NameComponent path[] = ncRef.to_name( name );
    ncRef.rebind(path, href);
   System.out.println("CalcyServer: Ready and waiting...");
   orb.run();
    }
    catch (Exception e)
    {
        System.out.println("CalcyServer: Some Error Has Occurred !");
        }
       System.out.println("\nCaclyServer: Exiting...");
    }
}

// CalcyClient.java

import CalcyApp.*;
import org.omg.CosNaming.*;
import org.omg.CosNaming.NamingContextPackage.*;
import org.omg.CORBA.*;
import java.util.Scanner;
import java.lang.*;

public class CalcyClient
{
  static CalcyInterface CalcyImpl;
  static int flag=1;
  static double x=0.0d;
  static double y=0.0d;
  public static void main(String args[])
   {
try{
                          ORB orb = ORB.init(args, null);
                          org.omg.CORBA.Object objRef =
                orb.resolve_initial_references("NameService");
                 NamingContextExt ncRef = NamingContextExtHelper.narrow(objRef); String
        name = "CalcyOperations";
                CalcyImpl = CalcyInterfaceHelper.narrow(ncRef.resolve_str(name));
                  System.out.print("CaclyClient: Obtained a handle on server object: \n\n");
                     System.out.println(CalcyImpl);


                Scanner sc=new Scanner(System.in);
               flag=1;
               do
                {
                   System.out.print("\nCaclyClient: Enter First Number: ");
                   x=sc.nextDouble();
                    System.out.print("\nCaclyClient: Enter Second Number: ");
                    y=sc.nextDouble();
                    System.out.println("-----------------------------------------------");
                    System.out.println("\n Addition\t= "+CalcyImpl.add(x,y));
                    System.out.println("\n Subtraction\t= "+CalcyImpl.subtract(x,y));
                    System.out.println("\n Multiplication\t= "+CalcyImpl.multiply(x,y));
                     System.out.println("\n Division\t= "+CalcyImpl.divide(x,y));
                     System.out.println("-----------------------------------------------");
                     System.out.println("Continue?[1:Yes|0:No]: ");
                      flag=sc.nextInt();
                 }
                 while (flag!=0);
                  CalcyImpl.shutdown();

          }

          catch (Exception e)

          {

        System.out.println("CalcyClient: Some Error Has Occurred !");

                  }

        }

        }
```

OUTPUT:



Command Prompt - java CalcyServer -ORBInitialPort 1050 -ORBInitialHost localhost

```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\depaul>cd desktop

C:\Users\depaul\Desktop>idlj -fall CalcyInterface.idl

C:\Users\depaul\Desktop>javac CalcyServer.java CalcyApp/*.java
Note: CalcyApp\CalcyInterfacePOA.java uses unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.

C:\Users\depaul\Desktop>javac CalcyClient.java CalcyApp/*.java
Note: CalcyApp\CalcyInterfacePOA.java uses unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.

C:\Users\depaul\Desktop>start orbd -ORBInitialPort 1050 -ORBInitialHost localhos
t

C:\Users\depaul\Desktop>  java CalcyServer -ORBInitialPort 1050 -ORBInitialHost
localhost
Exception in thread "main" java.lang.NoClassDefFoundError: CalcyServer

C:\Users\depaul\Desktop>set classpath=%classpath%;.;

C:\Users\depaul\Desktop>  java CalcyServer -ORBInitialPort 1050 -ORBInitialHost
localhost
CalcyServer: Ready and waiting...
```



Command Prompt - java CalcyClient -ORBInitialPort 1050 -ORBInitialHost localhost

```
C:\Users\depaul\Desktop>java CalcyClient -ORBInitialPort 1050 -ORBInitialHost lo
calhost
CaclyClient: Obtained a handle on server object:

IOR:000000000000002049444c3a43616c63794170702f43616c6379496e746572666163653a312e
3000000000010000000000000086000102000000000d3139322e3136382e312e31350000c0170000
0031afabcb000000002080202817000000010000000000000001000000008526f6f74504f41000000
00080000000100000000140000000000000200000001000000200000000000000100010000000020501
000100010020000101090000000100010100000000260000000200002

CaclyClient: Enter First Number: 3

CaclyClient: Enter Second Number: 5
-------------------------------------------------

 Addition       = 8.0

 Subtraction    = -2.0

 Multiplication = 15.0

 Division       = 0.6
-------------------------------------------------
Continue?[1:Yes|0:No]:
```