

# **ENTERPRISE JAVA PROGRAMMING**



**GOVT. COLLEGE THALASSERY**  
**KANNUR UNIVERSITY**

**CERTIFICATE**

This is to certify that this is the bonafide record of the original work done by Ms/Mr. .... Reg no..... of Sixth semester BCA in the programming in ENTERPRISE JAVA PROGRAMMING lab during the year 2018-19.

Head of the Department

Lecture in Charge

Submitted for the university Practical Examination held on .....

Examiner 1:

Examiner 2:

SERIAL NO.	PROGRAMS	PAGE NO.
1.	JDBC program to insert, Delete and Update records into Employee table.	
2.	JDBC program to Implement the record scrolling functions	
3.	JDBC program to Implement bank transaction	
4.	JDBC program to display database metadata	
5.	JDBC program to display Resultset metadata	
6.	RMI program for Complex number operation	
7.	RMI program for matrix addition and subtraction	
8.	RMI program for Bank operation	
9.	Servlet program that displays request information such as protocol etc	
10.	AtmServlet to handle bank operations	
11.	Servlet program that displays the contents of the file, specified by the user.	
12.	Servlet program that display student details	
13.	Session handling servlet that displays total number of visits to that page	

# PROGRAM 1

---

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

PROGRAM:

```
import java.io.*;

import java.sql.*;

public class Jdbcpgm1

{

public static void main(String[] args)

{

ResultSet rs;

ResultSetMetaData rm;

String eno, ename, salary;

int ch, n;

try

{

    Class.forName("com.mysql.jdbc.Driver");

    Connection con=DriverManager.getConnection( "jdbc:mysql://localhost:3306/Db1","root","gct2018");

    Statement stmt = con.createStatement();

    InputStreamReader isr=new InputStreamReader(System.in);

    BufferedReader br = new BufferedReader(isr);

    while(true)

    {

        System.out.println("1. Select");
```

```
System.out.println("2. Insert");

System.out.println("3. Update");

System.out.println("4. Delete");

System.out.println("5. Exit");

System.out.print("Enter your choice:");

ch = Integer.parseInt(br.readLine());

switch(ch)
{
case 1:

        rs = stmt.executeQuery("select * from emp");

        rm = rs.getMetaData();

        n= rm.getColumnCount();

        while(rs.next())

        {

                for(int i=1;i<=n;i++)

                        System.out.print(rs.getString(i) +"\t\t");

                System.out.println();

        }

        break;

case 2:

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

        eno = br.readLine();

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

        ename = br.readLine();

        System.out.println("Enter Salary");
```

```
salary = br.readLine();  
  
stmt.execute("insert into emp values("+eno+", '"+ename+"', "+salary+"");  
  
System.out.println("1 Record inserted");  
  
break;
```

case 3:

```
System.out.println("Enter E.No. to Edit :");  
  
eno = br.readLine();  
  
System.out.println("Enter Name");  
  
ename = br.readLine();  
  
System.out.println("Enter Salary");  
  
salary = br.readLine();
```

```
stmt.execute("update emp set ename='"+ename+"', salary= "+salary+" where eno  
='"+eno+"'");
```

```
System.out.println("1 Record Updated");  
  
break;
```

case 4:

```
System.out.println("Enter E.No. to Delete :");  
  
eno = br.readLine();  
  
stmt.execute("delete from emp where eno ='"+eno+"'");  
  
System.out.println("Record Deleted");  
  
break;
```

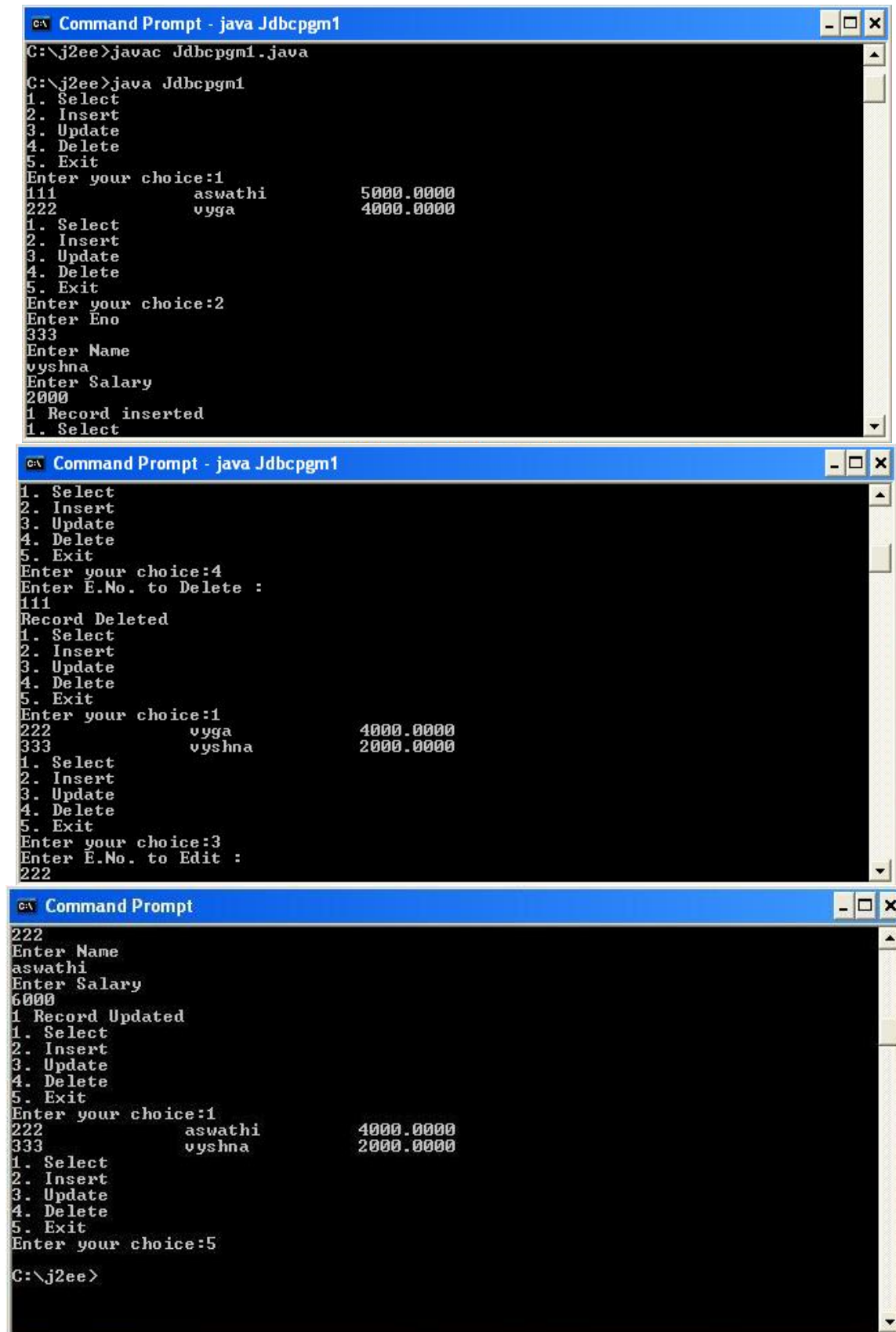
case 5:

```
br.close();  
  
con.close();  
  
System.exit(0);
```

## ENTERPRISE JAVA PROGRAMMING

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

### OUTPUT



The image shows three sequential screenshots of a Java application running in a Windows Command Prompt window. The window title is "Command Prompt - java Jdbcpgm1".

**First Screenshot:** The user has executed `javac Jdbcpgm1.java` and `java Jdbcpgm1`. The application displays a menu with options 1. Select, 2. Insert, 3. Update, 4. Delete, and 5. Exit. The user enters choice 1. The application displays two records:

E.No.	Name	Salary
111	aswathi	5000.0000
222	vyga	4000.0000

The user enters choice 2. The application prompts for "Enter E.No." and the user enters 333. It then prompts for "Enter Name" (vyshna) and "Enter Salary" (2000). It displays "1 Record inserted" and returns to the menu.

**Second Screenshot:** The user enters choice 4. The application prompts for "Enter E.No. to Delete :". The user enters 111. The application displays "Record Deleted" and returns to the menu.

**Third Screenshot:** The user enters choice 1. The application displays the updated records:

E.No.	Name	Salary
222	vyga	4000.0000
333	vyshna	2000.0000

The user enters choice 3. The application prompts for "Enter E.No. to Edit :". The user enters 222. It then prompts for "Enter Name" (aswathi) and "Enter Salary" (6000). It displays "1 Record Updated" and returns to the menu.

The user enters choice 5. The application displays the final records:

E.No.	Name	Salary
222	aswathi	4000.0000
333	vyshna	2000.0000

The user enters choice 5. The application displays "Enter your choice:5" and then the prompt `C:\j2ee>`.



## PROGRAM 2

---

**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.*;

class Jdbcpgm2
{
    public static void main(String args[])
    {
        try
        {
            Class.forName("com.mysql.jdbc.Driver");

            Connection con=DriverManager.getConnection( "jdbc:mysql://localhost:3306/Db1","root","gct2018");

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

            ResultSet rs=stmt.executeQuery("select * from student");

            System.out.println("Print records from bottom to top");

            rs.afterLast();

            while(rs.previous())
            {
                System.out.println(rs.getString(1)+" "+rs.getInt(2)+" "+rs.getString(3));
            }

            System.out.println("To print 3rd record data using absolute(3):");

            rs.absolute(3);

            System.out.println(rs.getString(1)+" "+rs.getInt(2)+" "+rs.getString(3));

            System.out.println("print 2nd record data using relative(-1)");
```

```

        rs.relative(-1);

        System.out.println(rs.getString(1)+" "+rs.getInt(2)+" "+rs.getString(3));

        System.out.println("print the first record after moving to first position with first(): ");

        rs.first();

        System.out.println(rs.getString(1)+" "+rs.getInt(2)+" "+rs.getString(3));

        System.out.println("print the last record after moving to last record using last():");

        rs.last();

        System.out.println(rs.getString(1)+" "+rs.getInt(2)+" "+rs.getString(3));

        System.out.println("Print records from top to bottom");

        rs.beforeFirst();

        while(rs.next())

        {

            System.out.println(rs.getString(1)+" "+rs.getInt(2)+" "+rs.getString(3));

        }

        rs.close();

        stmt.close();

        con.close();

    }

    catch(Exception e)

    {

        System.out.println(e);

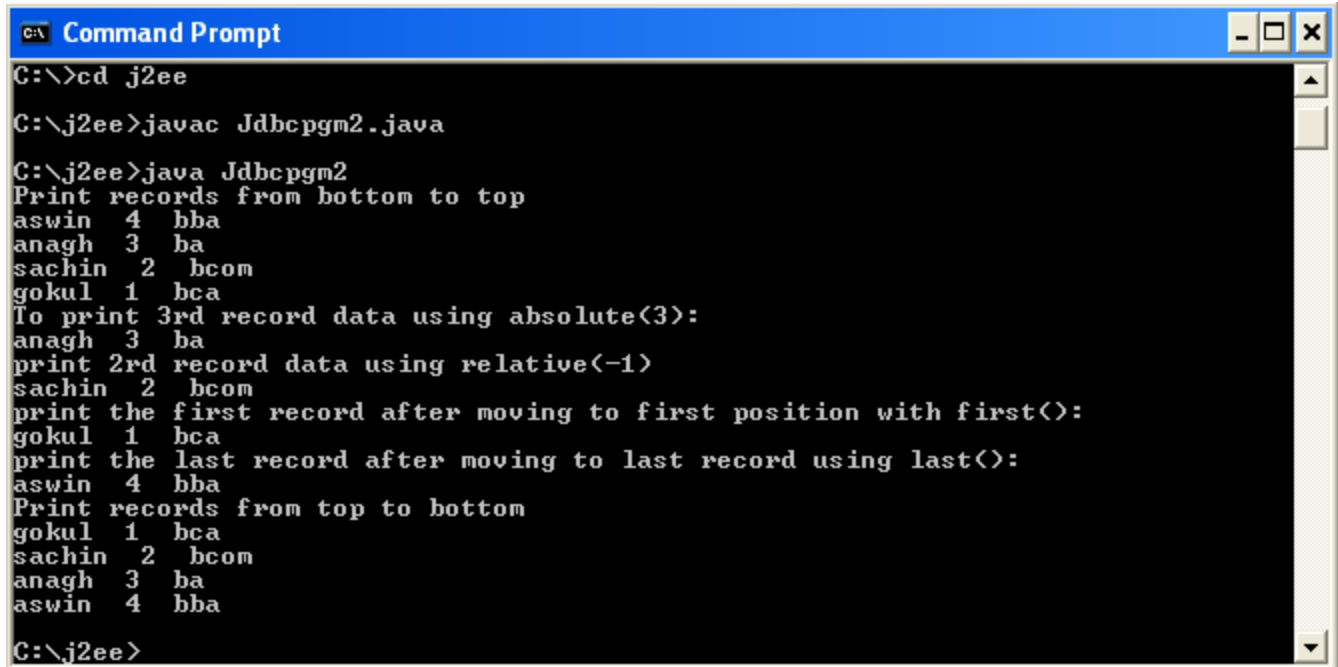
    }

}

}

```

**OUTPUT:**



```
C:\>cd j2ee
C:\j2ee>javac Jdbcpgm2.java
C:\j2ee>java Jdbcpgm2
Print records from bottom to top
aswin 4 bba
anagh 3 ba
sachin 2 bcom
gokul 1 bca
To print 3rd record data using absolute(3):
anagh 3 ba
print 2rd record data using relative(-1)
sachin 2 bcom
print the first record after moving to first position with first():
gokul 1 bca
print the last record after moving to last record using last():
aswin 4 bba
Print records from top to bottom
gokul 1 bca
sachin 2 bcom
anagh 3 ba
aswin 4 bba
C:\j2ee>
```

## PROGRAM 3

---

**AIM:** Create Bank table with fields `sacc_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.io.*;

import java.sql.*;

class Jdbcpgm3

{

public static void main(String[] args)

{

ResultSet rs;

int ch;

try

{

    Class.forName("com.mysql.jdbc.Driver");

    Connection con=DriverManager.getConnection( "jdbc:mysql://localhost:3306/Db1","root","gct2018");

    Statement stmt = con.createStatement();

    InputStreamReader isr=new InputStreamReader(System.in);

    BufferedReader br = new BufferedReader(isr);

    while(true)

    {

        System.out.println("Choose Option");

        System.out.println("1. Deposit");
```

```
System.out.println("2. Withdraw");

System.out.println("3. Trasaction");

System.out.println("4. Balance");

System.out.println("5. Exit");

ch = Integer.parseInt(br.readLine());

switch(ch)
{
case 1:

        System.out.println("Enter the account number:");

        String acc=br.readLine();

        System.out.println("Enter the amount to be deposited:");

        float amt=Float.parseFloat(br.readLine());

        rs=stmt.executeQuery("select balance from bank where acc_no="+acc+"");

        rs.next();

        float b=Float.parseFloat(rs.getString(1));

        float new_amt=amt+b;

        stmt.execute("update bank set balance="+new_amt+" where acc_no="+acc+"");

        break;

case 2:

        System.out.println("Enter the account number:");

        String acc1=br.readLine();

        System.out.println("Enter the amount to be deposited:");

        float amt1=Float.parseFloat(br.readLine());

        rs=stmt.executeQuery("select balance from bank where acc_no="+acc1+"");
```

```
rs.next();

float b1=Float.parseFloat(rs.getString(1));

float new_amt1=b1-amt1;

stmt.execute("update bank set balance="+new_amt1+" where acc_no="+acc1+"");

break;
```

case 3:

```
System.out.println("Enter the account number:");

String acc2=br.readLine();

System.out.println("Enter the account number do you want to transact :");

String acc3=br.readLine();

System.out.println("Enter the amount to be transacted:");

float amt2=Float.parseFloat(br.readLine());

rs=stmt.executeQuery("select balance from bank where acc_no="+acc2+"");

rs.next();

float b2=Float.parseFloat(rs.getString(1));

float new_amt2=b2-amt2;

stmt.execute("update bank set balance="+new_amt2+" where acc_no="+acc2+"");

rs=stmt.executeQuery("select balance from bank where acc_no="+acc3+"");

rs.next();

float b3=Float.parseFloat(rs.getString(1));

float new_amt3=amt2+b3;

stmt.execute("update bank set balance="+new_amt3+" where acc_no="+acc3+"");

break;
```

case 4:

```
        System.out.println("Enter the account no:");

        String n=br.readLine();

        rs=stmt.executeQuery("select balance from bank where acc_no="+n+"");

        while(rs.next())

            System.out.println("Balance:"+rs.getString(1));

        break;

    case 5:

        stmt.close();

        con.close();

        System.exit(0);

    }

}

}

catch(Exception e)

{

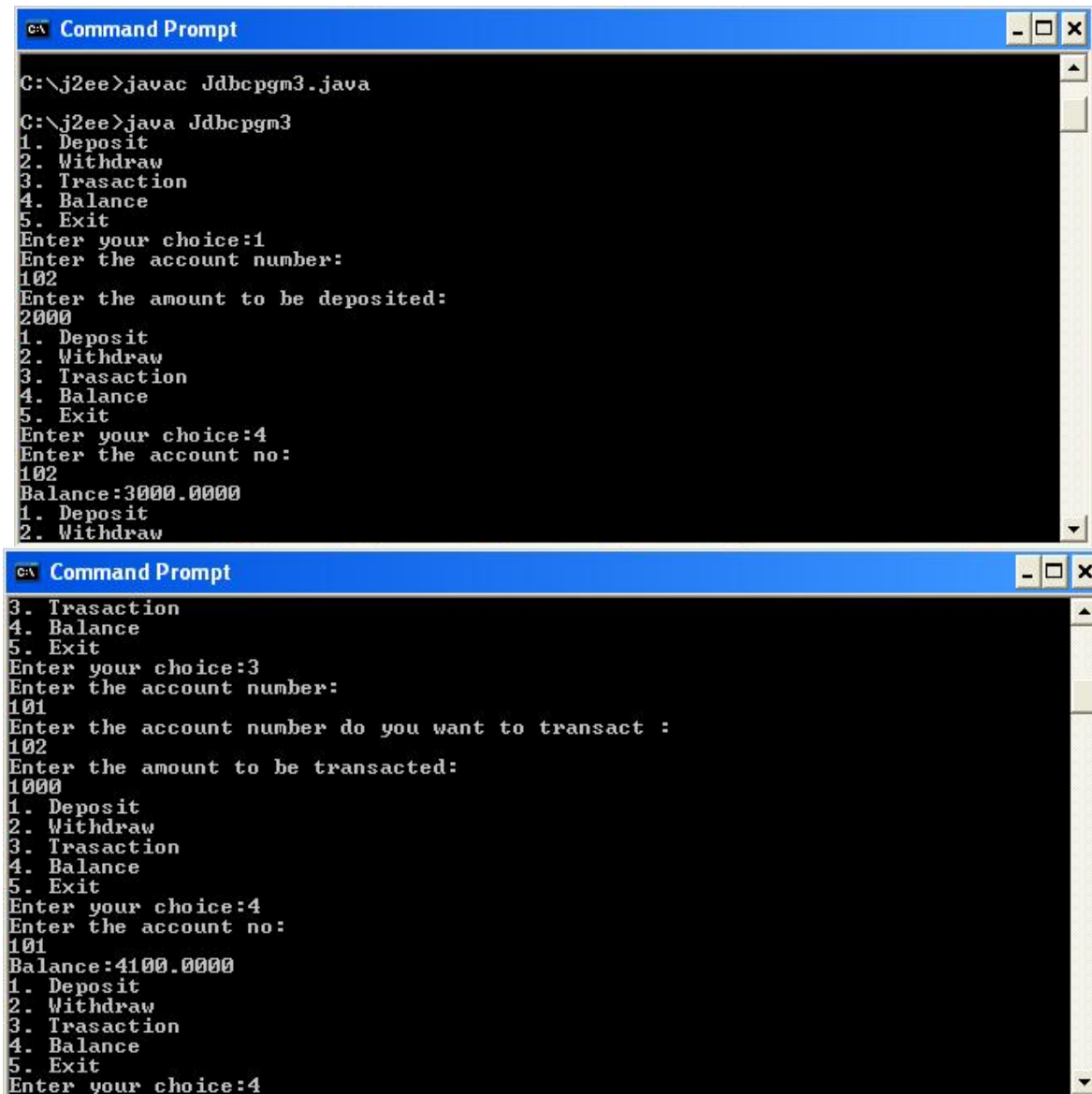
    System.out.println("Error :"+e.getMessage());

}

}

}
```

OUTPUT:



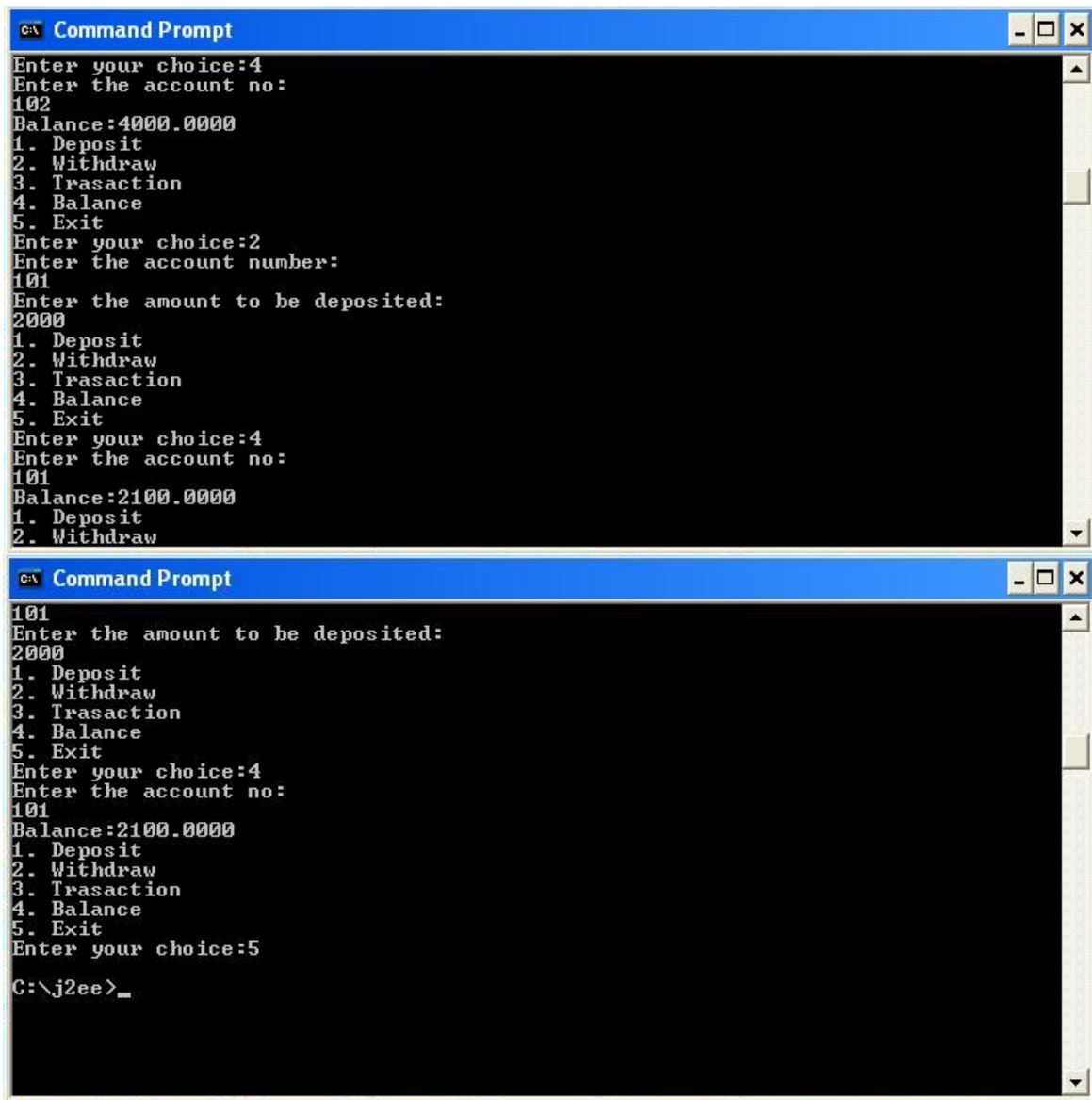
```

C:\j2ee>javac Jdbcpgm3.java

C:\j2ee>java Jdbcpgm3
1. Deposit
2. Withdraw
3. Trasaction
4. Balance
5. Exit
Enter your choice:1
Enter the account number:
102
Enter the amount to be deposited:
2000
1. Deposit
2. Withdraw
3. Trasaction
4. Balance
5. Exit
Enter your choice:4
Enter the account no:
102
Balance:3000.0000
1. Deposit
2. Withdraw

C:\j2ee>
3. Trasaction
4. Balance
5. Exit
Enter your choice:3
Enter the account number:
101
Enter the account number do you want to transact :
102
Enter the amount to be transacted:
1000
1. Deposit
2. Withdraw
3. Trasaction
4. Balance
5. Exit
Enter your choice:4
Enter the account no:
101
Balance:4100.0000
1. Deposit
2. Withdraw
3. Trasaction
4. Balance
5. Exit
Enter your choice:4
  
```





```
C:\> Command Prompt
Enter your choice:4
Enter the account no:
102
Balance:4000.0000
1. Deposit
2. Withdraw
3. Trasaction
4. Balance
5. Exit
Enter your choice:2
Enter the account number:
101
Enter the amount to be deposited:
2000
1. Deposit
2. Withdraw
3. Trasaction
4. Balance
5. Exit
Enter your choice:4
Enter the account no:
101
Balance:2100.0000
1. Deposit
2. Withdraw

C:\> Command Prompt
101
Enter the amount to be deposited:
2000
1. Deposit
2. Withdraw
3. Trasaction
4. Balance
5. Exit
Enter your choice:4
Enter the account no:
101
Balance:2100.0000
1. Deposit
2. Withdraw
3. Trasaction
4. Balance
5. Exit
Enter your choice:5
C:\j2ee>_
```

# PROGRAM 4

---

**AIM: JDBC program to display database metadata.**

PROGRAM:

```
import java.sql.*;

class Jdbcpgm4
{
    public static void main(String args[])
    {
        try
        {
            Class.forName("com.mysql.jdbc.Driver");

            Connection con=DriverManager.getConnection( "jdbc:mysql://localhost:3306/Db1","root","gct2018");

            DatabaseMetaData dbmd=con.getMetaData();

            System.out.println("Driver Name: "+dbmd.getDriverName());

            System.out.println("Driver Version: "+dbmd.getDriverVersion());

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

            System.out.println("Database Product Name: "+dbmd.getDatabaseProductName());

            System.out.println("Database Product Version: "+dbmd.getDatabaseProductVersion());

            con.close();

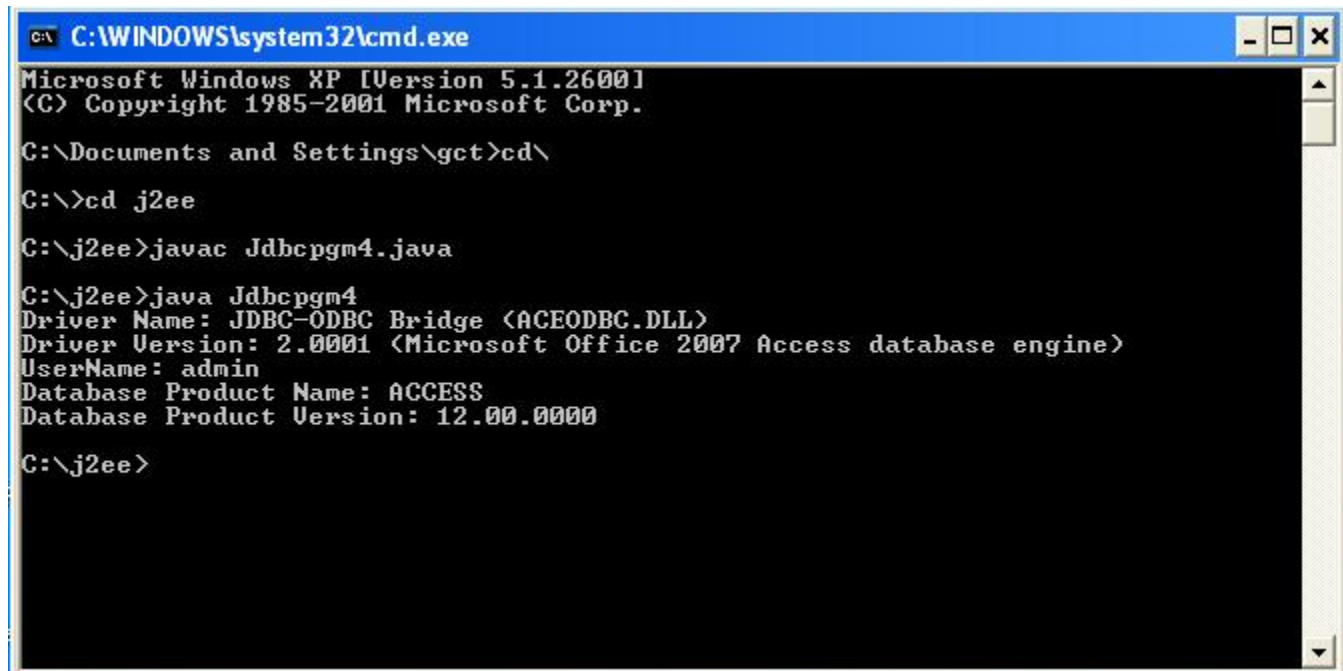
        }

        catch(Exception e)
        {

            System.out.println(e);
        }
    }
}
```

```
    }  
    }  
}
```

**OUTPUT:**



```
C:\WINDOWS\system32\cmd.exe  
Microsoft Windows XP [Version 5.1.2600]  
(C) Copyright 1985-2001 Microsoft Corp.  
  
C:\Documents and Settings\gct>cd\  
C:\>cd j2ee  
C:\j2ee>javac Jdbcpgm4.java  
C:\j2ee>java Jdbcpgm4  
Driver Name: JDBC-ODBC Bridge (ACEODBC.DLL)  
Driver Version: 2.0001 (Microsoft Office 2007 Access database engine)  
UserName: admin  
Database Product Name: ACCESS  
Database Product Version: 12.00.0000  
C:\j2ee>
```

# PROGRAM 5

---

**AIM: JDBC program to display Resultset metadata.**

PROGRAM:

```
import java.sql.*;

class Jdbcpgm5
{
    public static void main(String args[])
    {
        try
        {
            Class.forName("com.mysql.jdbc.Driver");

            Connection con=DriverManager.getConnection( "jdbc:mysql://localhost:3306/Db1","root","gct2018");

            PreparedStatement ps=con.prepareStatement("select * from student");

            ResultSet rs=ps.executeQuery();

            ResultSetMetaData rsmd=rs.getMetaData();

            System.out.println("Total columns: "+rsmd.getColumnCount());

            System.out.println("Column Name of 1st column: "+rsmd.getColumnName(1));

            System.out.println("Column Type Name of 1st column: "+rsmd.getColumnTypeName(1));

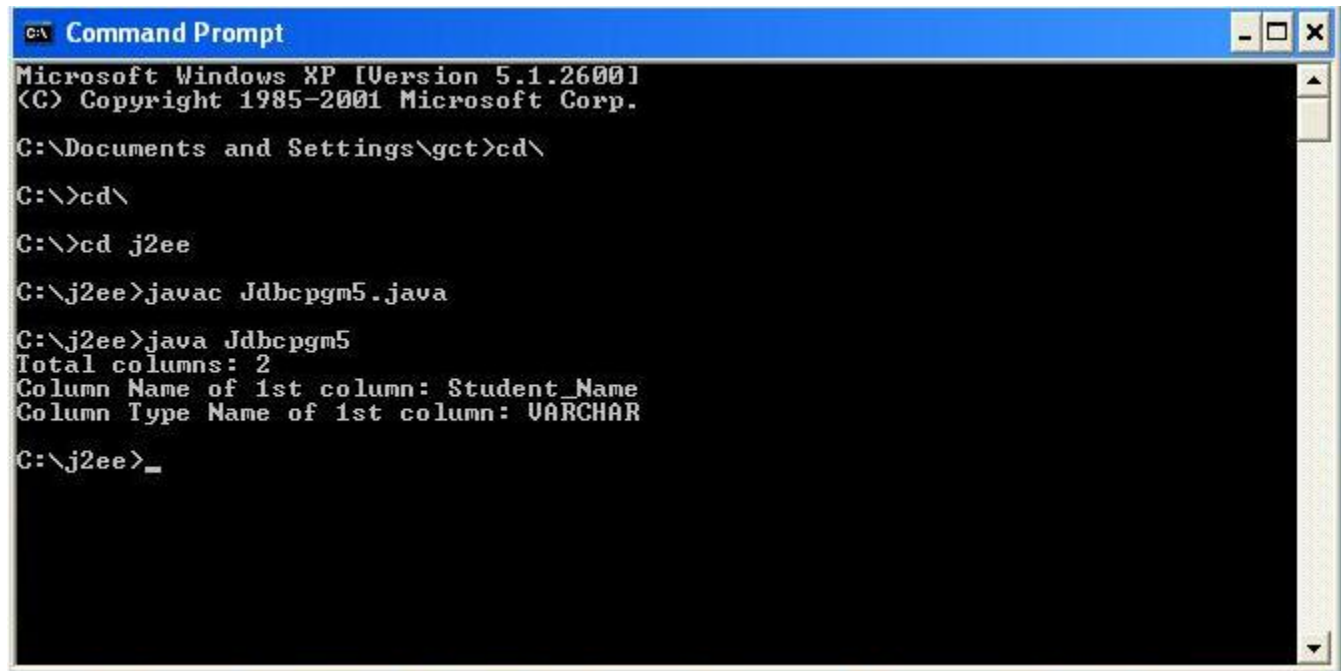
            con.close();
        }

        catch(Exception e)
        {

            System.out.println(e);
        }
    }
}
```

```
    }  
}  
}
```

**OUTPUT:**



```
C:\ Command Prompt  
Microsoft Windows XP [Version 5.1.2600]  
(C) Copyright 1985-2001 Microsoft Corp.  
  
C:\Documents and Settings\gct>cd\  
C:\>cd\  
C:\>cd j2ee  
C:\j2ee>javac Jdbcpgm5.java  
C:\j2ee>java Jdbcpgm5  
Total columns: 2  
Column Name of 1st column: Student_Name  
Column Type Name of 1st column: VARCHAR  
C:\j2ee>_
```

## PROGRAM 6

---

**Aim: RMI program for Complex number operation.**

PROGRAM:

```
//Complex.java

import java.rmi.*;

import java.rmi.server.*;

import java.io.*;

public class Complex implements Serializable
{
    int real,imag;

    public Complex(int a,int b)
    {
        real=a;
        imag=b;
    }
}

// ComplexI.java

import java.rmi.*;

import java.rmi.server.*;

public interface ComplexI extends Remote
{
    public Complex add(Complex c1,Complex c2)throws Exception;

    public Complex subtract(Complex c1,Complex c2)throws Exception;
```

## ENTERPRISE JAVA PROGRAMMING

```
        public Complex multiply(Complex c1,Complex c2)throws Exception;
    }

//ComplexC.java

import java.rmi.*;

import java.rmi.server.*;

import java.io.*;

public class ComplexC extends UnicastRemoteObject implements ComplexI
{
    Complex cs;

    public ComplexC()throws RemoteException,IOException
    {
        cs=new Complex(0,0);
    }

    public Complex add(Complex c1,Complex c2)throws Exception
    {
        cs.real=c1.real+c2.real;

        cs.imag=c1.imag+c2.imag;

        return cs;
    }

    public Complex subtract(Complex c1,Complex c2)throws Exception
    {
        cs.real=c1.real-c2.real;

        cs.imag=c1.imag-c2.imag;

        return cs;
    }
}
```

## ENTERPRISE JAVA PROGRAMMING

```
public Complex multiply(Complex c1,Complex 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;
}
}
```

//Server.java

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

//Client.java



## ENTERPRISE JAVA PROGRAMMING

```
import java.rmi.*;

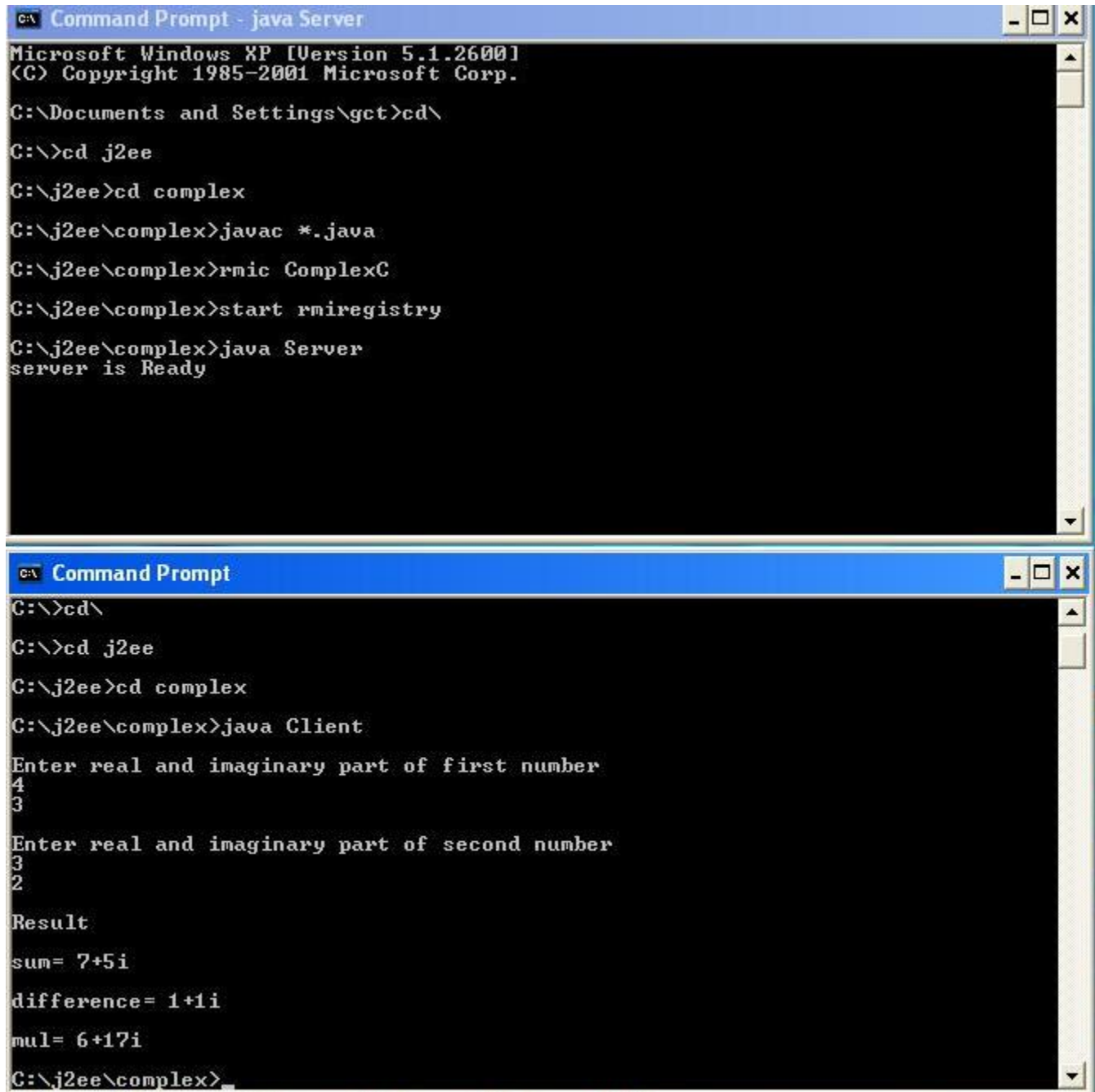
import java.io.*;

public class Client
{
    public static void main(String args[])
    {
        try
        {
            Complex com=(Complex)Naming.lookup("//127.0.0.1:1099/Comp");
            InputStreamReader isr=new InputStreamReader(System.in);
            BufferedReader br=new BufferedReader(isr);
            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());
            Complex cs1=new Complex(r1,i1);
            Complex cs2=new Complex(r2,i2);
            Complex cs3=new Complex(0,0);
            System.out.println("\nResult");
            cs3=com.add(cs1,cs2);
            System.out.println("\nsum= "+cs3.real+"+"+cs3.imag+"i");
            cs3=com.subtract(cs1,cs2);
            System.out.println("\ndifference= "+cs3.real+"+"+cs3.imag+"i");
```

## ENTERPRISE JAVA PROGRAMMING

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

## OUTPUT:



The image displays two screenshots of a Windows Command Prompt window. The top screenshot shows the process of starting a Java server. The bottom screenshot shows the process of running a Java client that interacts with the server.

```
C:\ Command Prompt - java Server
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\gct>cd\
C:\>cd j2ee
C:\j2ee>cd complex
C:\j2ee\complex>javac *.java
C:\j2ee\complex>rmic ComplexC
C:\j2ee\complex>start rmiregistry
C:\j2ee\complex>java Server
server is Ready
```

```
C:\ Command Prompt
C:\>cd\
C:\>cd j2ee
C:\j2ee>cd complex
C:\j2ee\complex>java Client
Enter real and imaginary part of first number
4
3
Enter real and imaginary part of second number
3
2
Result
sum= 7+5i
difference= 1+1i
mul= 6+17i
C:\j2ee\complex>
```

# PROGRAM 7

---

**AIM: RMI program for matrix addition and subtraction.**

PROGRAM:

```
//MatrixI.java
```

```
import java.rmi.*;
```

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

```
public interface MatrixI 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;
```

```
}
```

```
//MatrixC.java
```

```
import java.rmi.*;
```

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

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

```
public class MatrixC extends UnicastRemoteObject implements MatrixI
```

```
{
```

```
    public MatrixC()throws RemoteException
```

```
    {
```

```
    }
```

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

```
    {
```

```
        int s[][]=new int[r][c];
```

```

        int i,j;

        String str="";

        for(i=0;i<r;i++)
        {

            for(j=0;j<c;j++)

            {

                s[i][j]=a[i][j]+b[i][j];

                str=str+"\t"+String.valueOf(s[i][j]);

            }

            str=str+"\n";

        }

        return str;

    }

    public String sub(int a[][],int b[][],int r,int c)throws RemoteException,IOException

    {

        int d[][]=new int[r][c];

        int i,j;

        String str="";

        for(i=0;i<r;i++)

        {

            for(j=0;j<c;j++)

            {

                d[i][j]=a[i][j]-b[i][j];

                str=str+"\t"+String.valueOf(d[i][j]);

            }

        }

    }

```

## ENTERPRISE JAVA PROGRAMMING

```
        str=str+"\n";
    }
    return str;
}
}

//Server.java

import java.rmi.*;

import java.rmi.server.*;

public class Server
{
    public static void main(String args[])
    {
        try
        {
            MatrixC ob=new MatrixC();

            Naming.rebind("Matrix",ob);

            System.out.println("server is Ready");

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

        }
    }
}
```

## ENTERPRISE JAVA PROGRAMMING

```
//Client.java

import java.rmi.*;

import java.io.*;

public class Client

{

    public static void main(String args[])

    {

        try

        {

            int i,j,r,c;

            String str=new String("");

            InputStreamReader isr=new InputStreamReader(System.in);

            BufferedReader br=new BufferedReader(isr);

            MatrixI ob=(MatrixI)Naming.lookup("//127.0.0.1:1099/Matrix");

            System.out.println("Enter number of rows and columns");

            r=Integer.parseInt(br.readLine());

            c=Integer.parseInt(br.readLine());

            int a[][]=new int[r][c];

            int b[][]=new int[r][c];

            System.out.println("enter values of first matrix");

            for(i=0;i<r;i++)

            {

                for(j=0;j<c;j++)

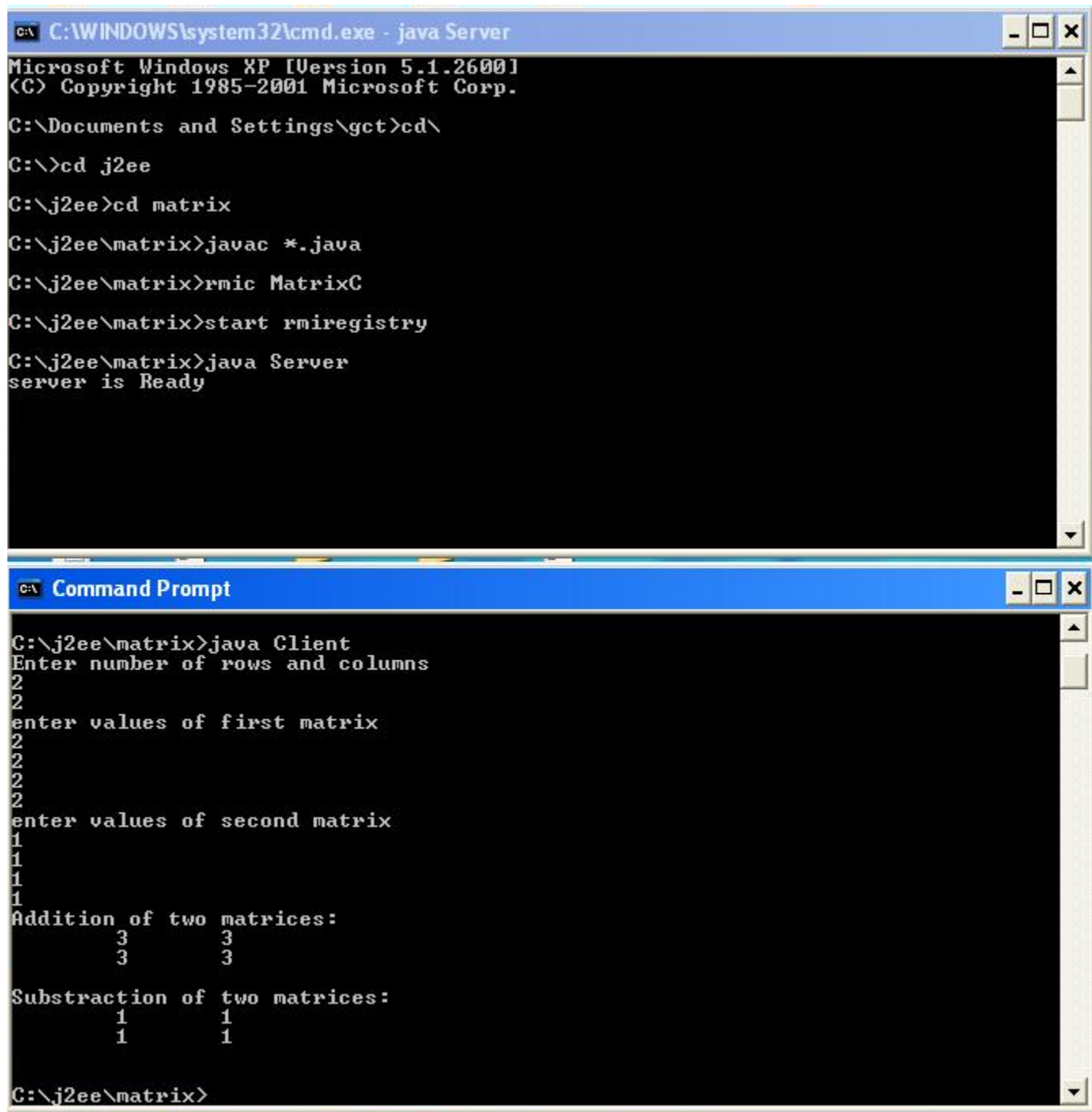
                {

                    a[i][j]=Integer.parseInt(br.readLine());
```

```
        }  
    }  
    System.out.println("enter values of second matrix");  
    for(i=0;i<r;i++)  
    {  
        for(j=0;j<c;j++)  
        {  
            b[i][j]=Integer.parseInt(br.readLine());  
        }  
    }  
    System.out.println("Addition of two matrices:");  
    str=(String)(ob.add(a,b,r,c));  
    System.out.println(str);  
    System.out.println("Substraction of two matrices:");  
    str=(String)(ob.sub(a,b,r,c));  
    System.out.println(str);  
  
    }  
    catch(Exception e)  
    {  
        System.out.println(e);  
    }  
}  
}
```



## OUTPUT:



```
C:\WINDOWS\system32\cmd.exe - java Server
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\gct>cd\
C:\>cd j2ee
C:\j2ee>cd matrix
C:\j2ee\matrix>javac *.java
C:\j2ee\matrix>rmic MatrixC
C:\j2ee\matrix>start rmiregistry
C:\j2ee\matrix>java Server
server is Ready

C:\j2ee\matrix>java Client
Enter number of rows and columns
2
2
enter values of first matrix
2
2
2
2
enter values of second matrix
1
1
1
1
Addition of two matrices:
    3    3
    3    3

Subtraction of two matrices:
    1    1
    1    1

C:\j2ee\matrix>
```

# PROGRAM 8

---

**AIM: RMI program for Bank operation.**

PROGRAM:

```
//BankI.java
```

```
import java.rmi.*;
```

```
public interface BankI extends Remote
```

```
{
```

```
    String getName()throws RemoteException;
```

```
    float balance()throws RemoteException;
```

```
    void withdraw(float amt)throws RemoteException;
```

```
    void deposit(float amt)throws RemoteException;
```

```
}
```

```
//BankC.java
```

```
import java.rmi.*;
```

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

```
public class BankC extends UnicastRemoteObject implements BankI
```

```
{
```

```
    String name;
```

```
    float bal;
```

```
    public BankC() throws RemoteException
```

```
    {
```

```
        super();
```

```
        name="John";
```

## ENTERPRISE JAVA PROGRAMMING

```
        bal=0;
    }
    public String getName() throws RemoteException
    {
        return name;
    }
    public float balance()throws RemoteException
    {
        return bal;
    }
    public void withdraw(float amt)throws RemoteException
    {
        bal=bal-amt;
    }
    public void deposit(float amt)throws RemoteException
    {
        bal=bal+amt;
    }
}

//Server.java

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

public class Server
{
```

## ENTERPRISE JAVA PROGRAMMING

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

        BankC ob=new BankC();

        Naming.rebind("rmi://127.0.0.1:1099/bank",ob);

        System.out.println("server is ready");

    }

    catch(Exception e)
    {

        System.out.println(e);

    }

}

//Client.java

import java.rmi.*;

import java.rmi.server.*;

import java.rmi.registry.*;

import java.io.*;

public class Client
{

    public static void main(String args[])
    {

        try{

            BankI stub=(BankI)Naming.lookup("rmi://127.0.0.1:1099/bank");

            while(true)
```

```

{
InputStreamReader isr=new InputStreamReader(System.in);
BufferedReader br=new BufferedReader(isr);
System.out.println("\n1.Account holder\n2.Deposit\n3.Withdraw\n4.balance\n5.Exit\n");
System.out.print("Enter your choice:");
int i=Integer.parseInt(br.readLine());
switch(i)
{
case 1:
    String s=stub.getName();
    System.out.println("Account Holder is "+s);
    break;
case 2:
    System.out.println("Enter the amount to be deposited");
    float a=Float.parseFloat(br.readLine());
    stub.deposit(a);
    float db=stub.balance();
    System.out.println("Balance="+db);
    break;
case 3:
    System.out.println("Enter the amount to be withdrawn");
    float w=Float.parseFloat(br.readLine());
    float wb=stub.balance();
    if(wb<w)
        System.out.println("Insuffient balance\n");

```

```
        else

            stub.withdraw(w);

            float wb1=stub.balance();

            System.out.println("Balance="+wb1);

            break;

        case 4:

            float b=stub.balance();

            System.out.println("Balance="+b);

            break;

        case 5:

            System.exit(0);

    }

}

catch(Exception e)

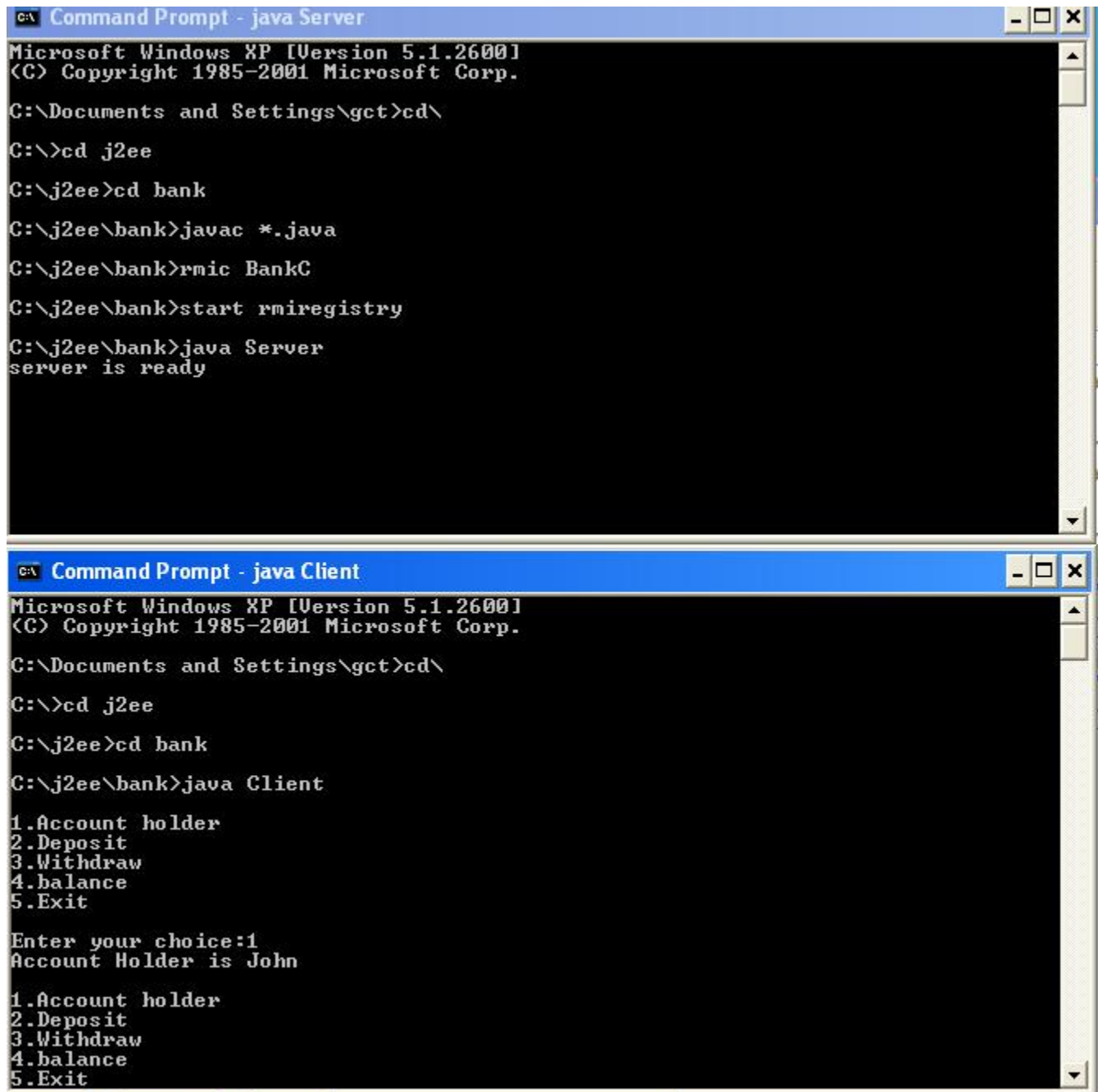
{

    System.out.println(e);

}

}
```

## OUTPUT:



The image displays two separate Command Prompt windows from a Windows XP environment. The top window, titled 'Command Prompt - java Server', shows the following sequence of commands and output: navigating to 'C:\j2ee\bank', compiling 'BankC.java' with 'javac \*.java', starting the 'rmiregistry' service, and finally running 'java Server', which outputs 'server is ready'. The bottom window, titled 'Command Prompt - java Client', shows the same directory navigation, followed by running 'java Client'. This produces a menu with five options: '1.Account holder', '2.Deposit', '3.Withdraw', '4.balance', and '5.Exit'. After selecting '1', the output is 'Account Holder is John', and the menu is repeated.

```
C:\> Command Prompt - java Server
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

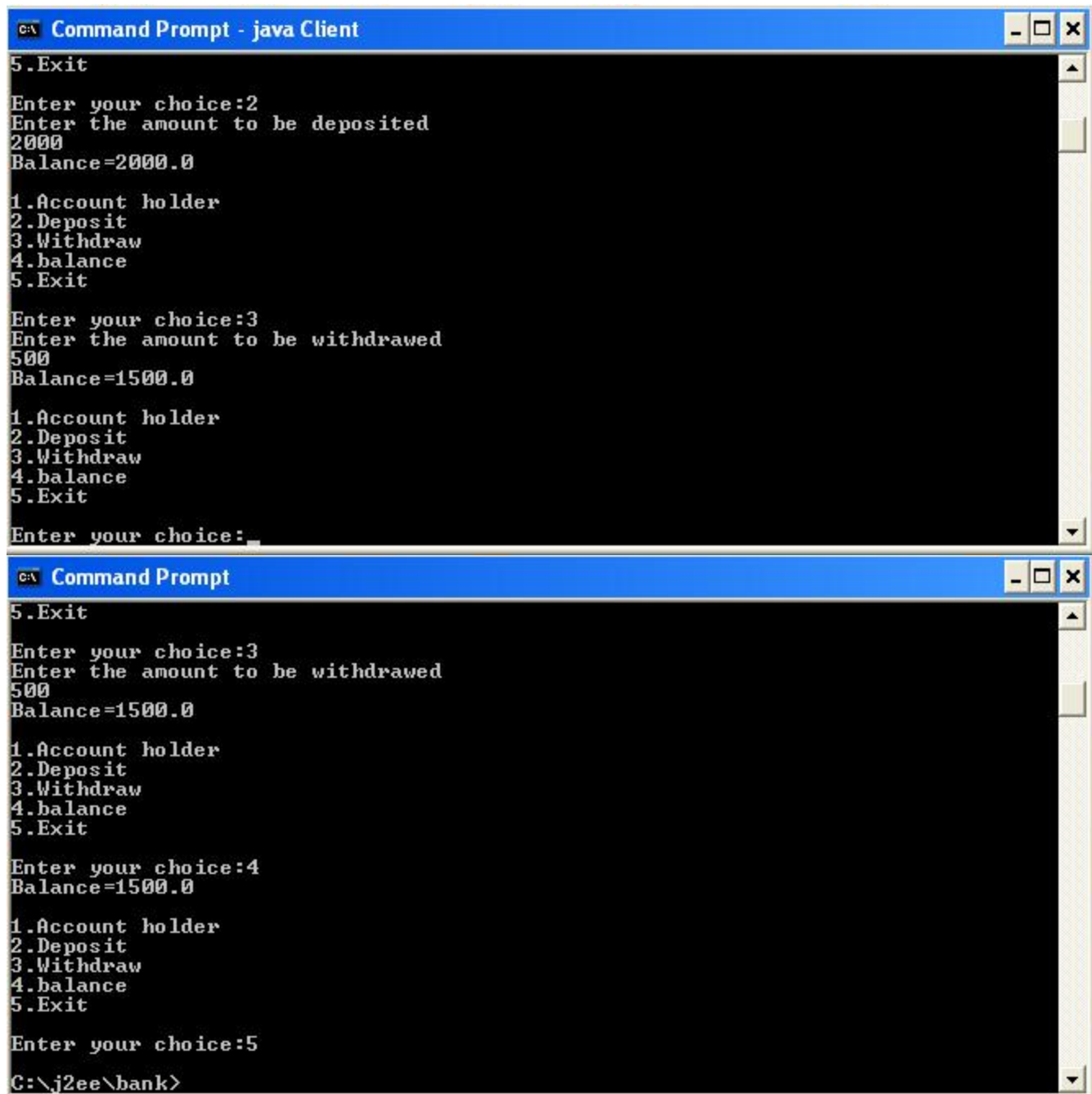
C:\Documents and Settings\gct>cd\
C:\>cd j2ee
C:\j2ee>cd bank
C:\j2ee\bank>javac *.java
C:\j2ee\bank>rmic BankC
C:\j2ee\bank>start rmiregistry
C:\j2ee\bank>java Server
server is ready

C:\> Command Prompt - java Client
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\gct>cd\
C:\>cd j2ee
C:\j2ee>cd bank
C:\j2ee\bank>java Client
1.Account holder
2.Deposit
3.Withdraw
4.balance
5.Exit

Enter your choice:1
Account Holder is John

1.Account holder
2.Deposit
3.Withdraw
4.balance
5.Exit
```



The image displays two sequential screenshots of a Java Client application running in a Windows Command Prompt window. The window title is "C:\ Command Prompt - java Client".

**Top Screenshot:**

```
5.Exit
Enter your choice:2
Enter the amount to be deposited
2000
Balance=2000.0

1.Account holder
2.Deposit
3.Withdraw
4.balance
5.Exit

Enter your choice:3
Enter the amount to be withdrawn
500
Balance=1500.0

1.Account holder
2.Deposit
3.Withdraw
4.balance
5.Exit

Enter your choice:_
```

**Bottom Screenshot:**

```
5.Exit

Enter your choice:3
Enter the amount to be withdrawn
500
Balance=1500.0

1.Account holder
2.Deposit
3.Withdraw
4.balance
5.Exit

Enter your choice:4
Balance=1500.0

1.Account holder
2.Deposit
3.Withdraw
4.balance
5.Exit

Enter your choice:5
C:\j2ee\bank>
```



## PROGRAM 9

---

**AIM: Servlet program that displays request information such as protocol, remote host name, server name, server port, Header names, specific headers, authentication type, scheme etc.**

PROGRAM:

```
import javax.servlet.http.*;

import javax.servlet.*;

import java.io.*;

public class ServletInfo extends HttpServlet

{

    public void doGet(HttpServletRequest req, HttpServletResponse res) throws
ServletException, IOException

    {

        res.setContentType("text/html");

        PrintWriter pw=res.getWriter();

        pw.println("<html><body>");

        pw.println("<b>"+req.getProtocol()+"<br>");

        pw.println("<b>"+req.getServerName()+"<br>");

        pw.println("<b>"+req.getServerPort()+"<br>");

        pw.println("<b>"+req.getRemoteHost()+"<br>");

        pw.println("<b>"+req.getHeader("Accept")+"<br>");

        pw.println("<b>"+req.getHeader("User-Agent")+"<br>");

        pw.println("<b>"+req.getAuthType()+"<br>");

        pw.println("<b>"+req.getScheme()+"<br>");

        pw.println("</body></html>");

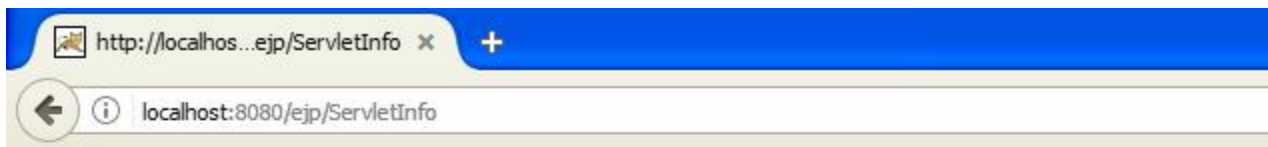
        pw.close();

    }

}
```

```
    }  
}  
//web.xml  
  
<servlet>  
    <servlet-name> ServletInfo </servlet-name>  
    <servlet-class> ServletInfo </servlet-class>  
  
</servlet>  
  
<servlet-mapping>  
    <servlet-name> ServletInfo </servlet-name>  
    <url-pattern>/ ServletInfo </url-pattern>  
  
</servlet-mapping>
```

**OUTPUT:**



**Protocol: HTTP/1.1**

**Server Name: localhost**

**Server port: 8080**

**Remote Host: 127.0.0.1**

**Header Name: text/html,application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8**

**Specific Header: Mozilla/5.0 (Windows NT 5.1; rv:52.0) Gecko/20100101 Firefox/52.0**

**Authentication Type: null**

**Scheme: http**

# PROGRAM 10

---

**AIM: AtmServlet to handle bank operations.**

```
import javax.servlet.http.*;

import javax.servlet.*;

import java.io.*;

public class AtmServlet extends HttpServlet

{

    Account act;

    public void init(ServletConfig conf) throws ServletException

    {

        super.init(conf);

        act=new Account();

        act.balance=0;

    }

    public void doGet(HttpServletRequest req,HttpServletResponse res)throws

ServletException,IOException

    {

        res.setContentType("text/html");

        PrintWriter pw=res.getWriter();

        pw.println("<html><body>");

        pw.println("Current balance="+act.balance+"<br>");

        pw.println("<form method=post action=AtmServlet>");

        pw.println("Amount:<input type=text name=amt>"+<br><br>");

        pw.println("<input type=submit value=Withdraw name=withdraw>");
```

```

        pw.println("<input type=submit value=Deposit name=deposit>");

        pw.println("</form>");

        pw.println("</body></html>");

    }

    public void doPost(HttpServletRequest req,HttpServletResponse res)throws
ServletException,IOException

    {

        res.setContentType("text/html");

        PrintWriter pw=res.getWriter();

        int amt=0;

        try

        {

            amt=Integer.parseInt(req.getParameter("amt"));

        }

        catch(NumberFormatException e)

        {

            pw.println("No amount is passed"+"<br>");

        }

        String s1=req.getParameter("withdraw");

        String s2=req.getParameter("deposit");

        if(s1!=null&&amt<act.balance)

        {

            act.balance=act.balance-amt;

        }

        if(s2!=null&&amt>0)

        {

```

```
        act.balance=act.balance+amt;

    }

    doGet(req,res);

}

class Account

{

    public int balance;

}

}
```

**//web.xml**

```
<servlet>

    <servlet-name> AtmServlet </servlet-name>

    <servlet-class> AtmServlet </servlet-class>

</servlet>

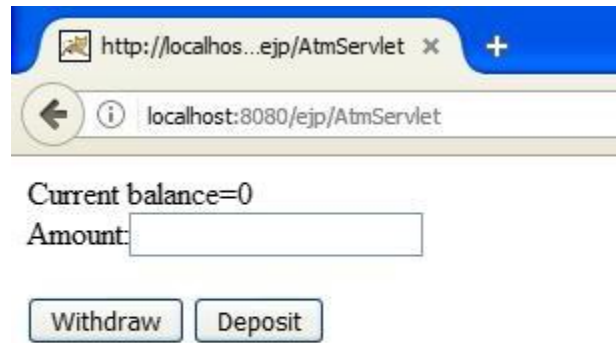
<servlet-mapping>

    <servlet-name> AtmServlet </servlet-name>

    <url-pattern>/ AtmServlet </url-pattern>

</servlet-mapping>
```

OUTPUT:

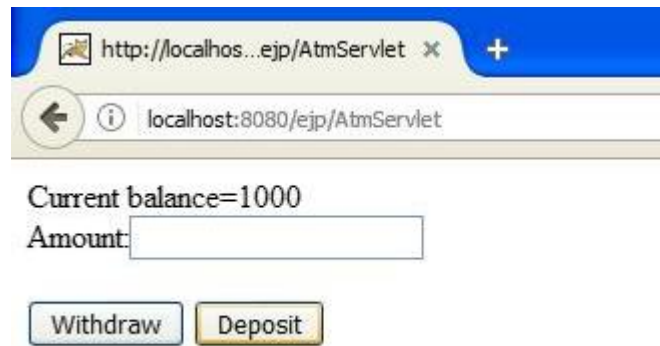


http://localhost:8080/ejp/AtmServlet

Current balance=0

Amount:

Withdraw Deposit

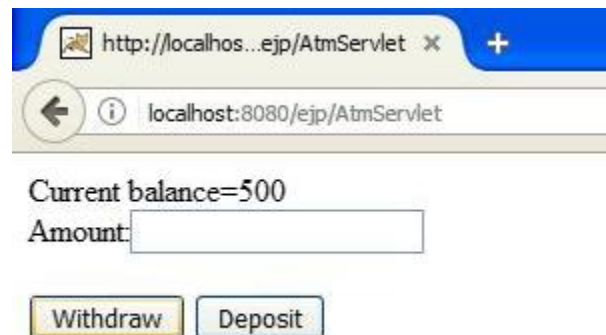


http://localhost:8080/ejp/AtmServlet

Current balance=1000

Amount:

Withdraw Deposit



http://localhost:8080/ejp/AtmServlet

Current balance=500

Amount:

Withdraw Deposit

# PROGRAM 11

---

**AIM:** Create an HTML form that reads a file name from the user. Write a servlet program that displays the contents of the file, specified by the user.

PROGRAM:

```
//FileServlet.html
```

```
<html>
```

```
<body>
```

```
<form method=get action="FileServlet">
```

```
User name:<input type="text" name="user">
```

```
<input type="file" name="filename">
```

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

```
</form>
```

```
</body>
```

```
</html>
```

```
//FileServlet.java
```

```
import javax.servlet.ServletException;
```

```
import javax.servlet.http.*;
```

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

```
public class FileServlet extends HttpServlet
```

```
{
```

```
    public void service(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException
```

```
    {
```

```
        res.setContentType("text/html");
```

```
        PrintWriter pw = res.getWriter();
```

## ENTERPRISE JAVA PROGRAMMING

```
String name = req.getParameter("filename");

BufferedReader br = new BufferedReader(new FileReader("c:/" + name));

String str;

while( (str = br.readLine()) != null )

{

    pw.println(str + "<BR>");

}

br.close();

pw.close();

}

}
```

**//web.xml**

```
<servlet>

    <servlet-name>FileServlet</servlet-name>

    <servlet-class>FileServlet</servlet-class>

</servlet>

<servlet-mapping>

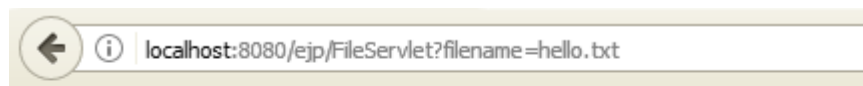
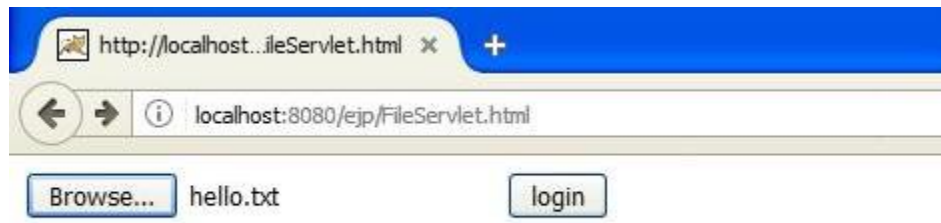
    <servlet-name>FileServlet</servlet-name>

    <url-pattern>/FileServlet</url-pattern>

</servlet-mapping>
```



**OUTPUT:**



Hello Servlet

# PROGRAM 12

---

**AIM: Create a html form to read student details such as Roll , Name ,Age ,Sex ,Qualification ,Percentage of mark ,write a servlet program that display same details**

**//Student.html**

```
<html>

<body>

    <form action="Student" method="GET">

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

        <br />

        Roll No: <input type="text" name="roll" />

        <br />

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

        <br />

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

        <br />

        Qualification:<input type="text" name="qualification">

        <br />

        Percetage of mark: <input type="text" name="percent" />

        <br />

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

    </form>

</body>

</html>
```

**//Student.java**

```
import java.io.*;

import javax.servlet.*;

import javax.servlet.http.*;

public class Student extends HttpServlet

{

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

{

    response.setContentType("text/html");

    PrintWriter out = response.getWriter();

    String title = "Using GET Method to Read Form Data";

    out.println( "<html>");

    out.println("<b>Name</b>: ");

    out.println(request.getParameter("name") );

    out.println("<b>Roll No</b>: ");

    out.println(request.getParameter("roll"));

    out.println("<b>Age</b>: ");

    out.println(request.getParameter("age") );

    out.println("<b>Sex</b>: ");

    out.println(request.getParameter("sex") );

    out.println("<b>Qualification</b>: ");

    out.println(request.getParameter("qualification"));

    out.println("<b>Percentage of mark:</b>: ");

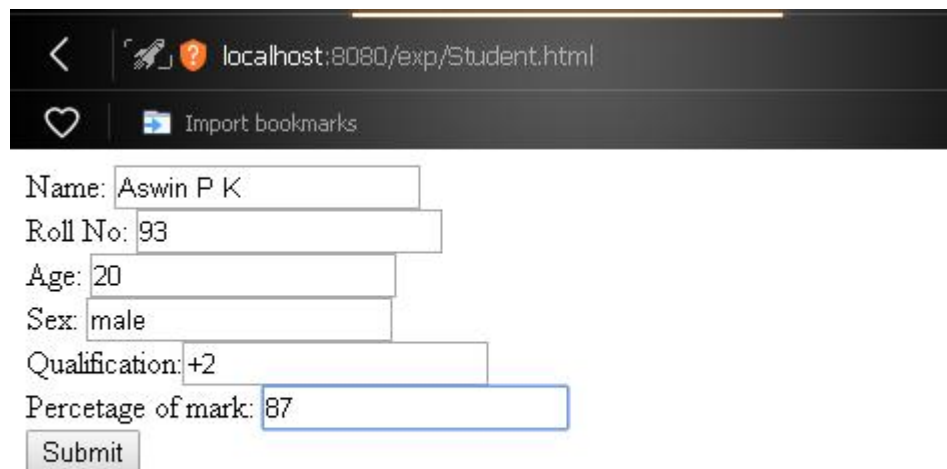
    out.println(request.getParameter("percent"));

    out.println("</body></html>");
```

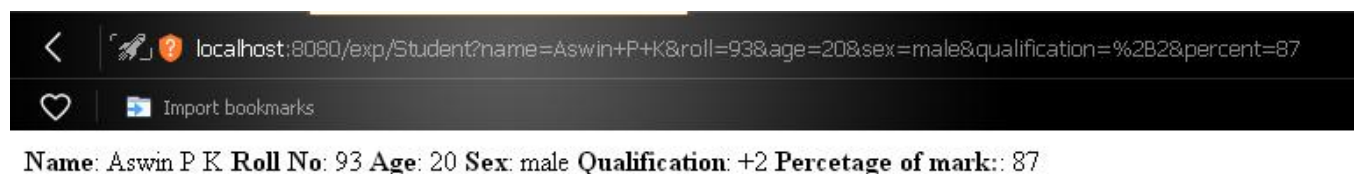
## ENTERPRISE JAVA PROGRAMMING

```
}  
}  
  
//web.xml  
  
<servlet>  
    <servlet-name> Student</servlet-name>  
    <servlet-class> Student </servlet-class>  
  
</servlet>  
  
<servlet-mapping>  
    <servlet-name> Student </servlet-name>  
    <url-pattern>/ Student </url-pattern>  
  
</servlet-mapping>
```

### OUTPUT:



A screenshot of a web browser window. The address bar shows 'localhost:8080/exp/Student.html'. Below the address bar is a navigation bar with a heart icon and 'Import bookmarks'. The main content area displays a form with the following fields and values: 'Name: Aswin P K', 'Roll No: 93', 'Age: 20', 'Sex: male', 'Qualification: +2', and 'Perctage of mark: 87'. A 'Submit' button is located at the bottom of the form.



A screenshot of a web browser window showing the output of the form submission. The address bar shows 'localhost:8080/exp/Student?name=Aswin+P+K&roll=93&age=20&sex=male&qualification=%2B2&percent=87'. Below the address bar is a navigation bar with a heart icon and 'Import bookmarks'. The main content area displays the output: 'Name: Aswin P K Roll No: 93 Age: 20 Sex: male Qualification: +2 Perctage of mark:: 87'.

# PROGRAM 13

---

**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())

        {

            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><b>Session Information</h2>");

        out.println("<br><b>Session info:"+session.getId());

        out.println("<br><b>Time of LastAccess:"+lastAccessTime);

        out.println("<br><b>User ID:"+userID);

        out.println("<br><b>Number of visits:"+visitCount);

    }

}

```

**//web.xml**

```

<servlet>

    <servlet-name> ServletTrack</servlet-name>

    <servlet-class> ServletTrack</servlet-class>

</servlet>

<servlet-mapping>

    <servlet-name> ServletTrack </servlet-name>

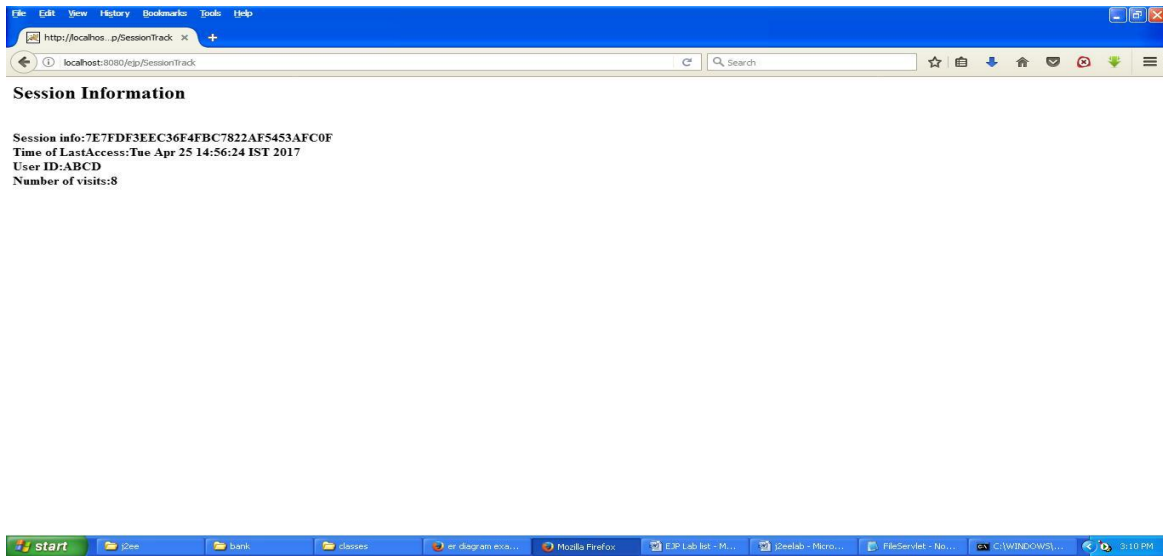
    <url-pattern>/ ServletTrack </url-pattern>

</servlet-mapping>

```

## ENTERPRISE JAVA PROGRAMMING

### OUTPUT:



# PROGRAM 14

---

**AIM: CORBA Program for Arithmetic Operation.**

**Arithmetic.idl:**

```
interface Arithmetic
{
    float add(in float a,in float b);
    float sub(in float a,in float b);
    float mul(in float a,in float b);
    float div(in float a,in float b);
};
```

**ArithmeticImp.java:**

```
public class ArithmeticImp extends _ArithmeticImplBase
{
    float c;
    public float add(float a,float b)
    {
        c=a+b;
        return c;
    }
    public float sub(float a,float b)
    {
        c=a-b;
        return c;
    }
}
```



```

    public float mul(float a,float b)
    {
        c=a*b;
        return c;
    }

    public float div(float a,float b)
    {
        c=a/b;
        return c;
    }
}

```

**Client.java:**

```

import org.omg.CORBA.*;
import org.omg.CosNaming.*;
import java.io.*;

public class Client
{
    public static void main(String arg[])
    {
        try{
            float a,b,c;

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

            ORB orb=ORB.init(arg,null);

            org.omg.CORBA.Object ob=orb.resolve_initial_references("NameService");

            NamingContext ctx=NamingContextHelper.narrow(ob);

```

```

        NameComponent nc=new NameComponent("Message","");
        NameComponent path[]={ nc};
        Arithmetic ar=ArithmeticHelper.narrow(ctx.resolve(path));
        System.out.println("Enter two numbers");
        a=Float.parseFloat(rd.readLine());
        b=Float.parseFloat(rd.readLine());
        c=ar.add(a,b);
        System.out.println("Sum="+c);
        c=ar.sub(a,b);
        System.out.println("Substract="+c);
        c=ar.mul(a,b);
        System.out.println("product="+c);
        c=ar.div(a,b);
        System.out.println("division="+c);
    }
    catch(Exception e)
    {}
}
}

```

**Server.java:**

```

import org.omg.CORBA.*;
import org.omg.CosNaming.*;

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

```

```
{
    try
    {
        ORB orb=ORB.init(arg,null);
        org.omg.CORBA.Object ob=orb.resolve_initial_references("NameService");
        NamingContext ctx=NamingContextHelper.narrow(ob);
        NameComponent nc=new NameComponent("Message","");
        NameComponent path[]={ nc};
        ArithmeticImp m=new ArithmeticImp();
        ctx.rebind(path,m);
        orb.run();
    }
    catch(Exception e)
    { }
}
}
```

**To compile:**

Idlj -fall -oldImplBase Arithmetic.idl

javac ArithmeticImp.java

javac Server.java

javac Client.java

**To run:**

1.orbd -ORBInitialPort 1050 -ORBInitialHost localhost&

2.java reg -ORBInitialPort 1050 -ORBInitialHost localhost&

3.java client -ORBInitialPort 1050 -ORBInitialHost localhost&

## OUTPUT

```

Command Prompt - orbd -ORBInitialPort 1050 -ORBInitialHost localhost
Microsoft Windows [Version 10.0.17134.590]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\DELL>e:

E:\>cd corbaexp

E:\corbaexp>Idlj -fall -oldImplBase Arithmetic.idl

E:\corbaexp>javac ArithmeticImp.java
Note: .\ArithmeticImplBase.java uses unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.

E:\corbaexp>javac -Xlint ArithmeticImp.java
ArithmeticImp.java:1: warning: [serial] serializable class ArithmeticImp has no definition of serialVersionUID
public class ArithmeticImp extends _ArithmeticImplBase
      ^
1 warning

E:\corbaexp>javac Client.java

E:\corbaexp>javac Server.java

E:\corbaexp>orbd -ORBInitialPort 1050 -ORBInitialHost localhost&

```

```

Command Prompt - java Server -ORBInitialPort 1050 -ORBInitialHost localhost
Microsoft Windows [Version 10.0.17134.590]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\DELL>e:

E:\>cd corbaexp

E:\corbaexp>java Server -ORBInitialPort 1050 -ORBInitialHost localhost&

```

```

Command Prompt
Microsoft Windows [Version 10.0.17134.590]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\DELL>e:

E:\>cd corbaexp

E:\corbaexp>java Client -ORBInitialPort 1050 -ORBInitialHost localhost&
Enter two numbers
5
10
Sum=15.0
Substract=-5.0
product=50.0
division=0.5

E:\corbaexp>

```

