GURUDEV ARTS & SCIENCE COLLEGE

MATHIL, PAYYANUR, KANNUR DIST.

(Affiliated to Kannur University)

BACHELOR OF COMPUTER APPLICATION

ENTERPRISE JAVA PROGRAMMING PRACTICAL RECORD

Name :

Class No :

University Reg. No :

GURUDEV ARTS & SCIENCE COLLEGE

MATHIL, PAYYANUR, KANNUR DIST.

(Affiliated to Kannur University)

DEPARTMENT OF COMPUTER SCIENCE

Name : ……..

Class No : …… University Reg. No : …….

Certified that this is the bonafide record of practicals done by .....……. of …… year ……..... in Gurudev Arts and Science College, Mathil for the year 20….. to 20…..

Lecturer in Charge Head Of Dept., Computer Science

Submitted for University Examinations 20…..

Examiners : 1.

2.

PROGRAM -1

JDBC program to insert, delete and update records into employee table.

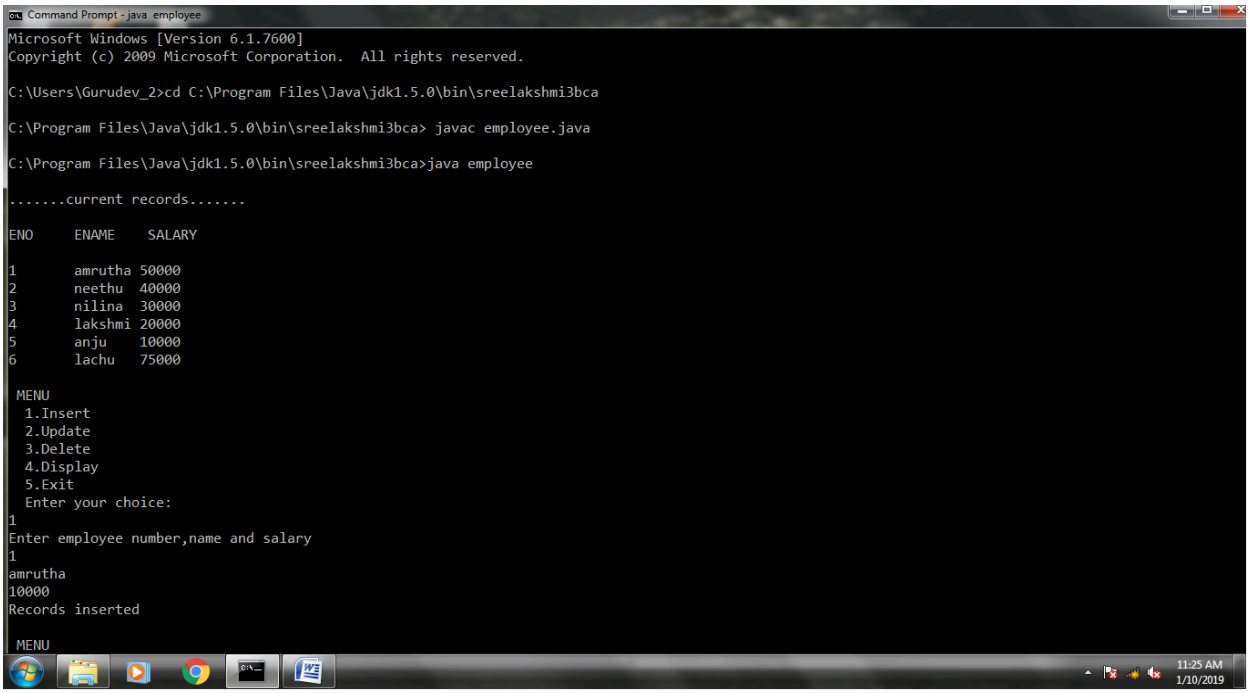
```
import java.sql.*;
import java.io.*;
public class employee
{
    static final String JDBC_DRIVER="com.mysql.jdbc.Driver";
    static final String DB_URL="jdbc:mysql://localhost/abcd";
    static final String USER="root";
    static final String PASS="gurudev";
    public static void main(String[] args)throws IOException
    {
       int ch,upc;
       int no,sal;
       String name;
       BufferedReader br=new BufferedReader(new
InputStreamReader(System.in));
     try
      {
         Class.forName("com.mysql.jdbc.Driver");
       }
         catch(ClassNotFoundException e)
    {
       System.out.println("Unable to load driver");
}
try
{
  Connection conn=DriverManager.getConnection(DB_URL,USER,PASS);
  Statement stmt=conn.createStatement();
  System.out.println("\n.......current records.......\n");
  System.out.println("ENO\tENAME \t SALARY \n");
  ResultSet rs=stmt.executeQuery("select * from employee");
  while(rs.next())
  {

System.out.println(rs.getInt("eno")+"\t"+rs.getString("ename")+"\t"+rs.getInt("sa
lary"));
  }
```

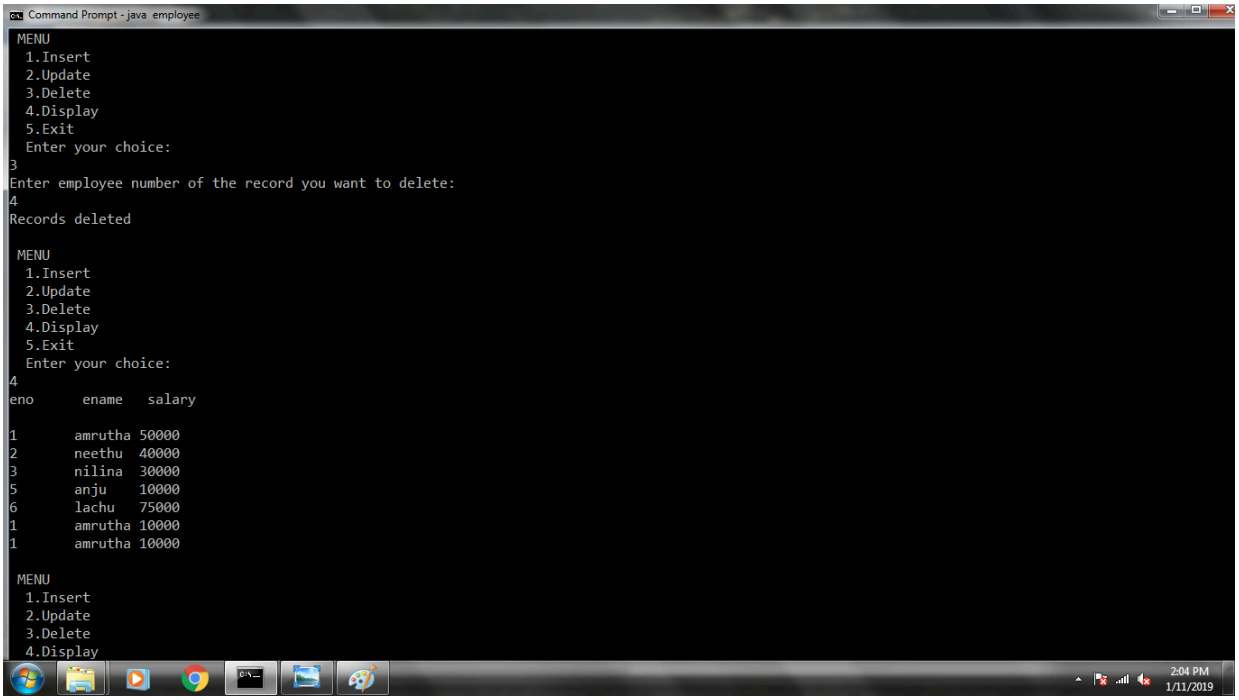
```
 do
{
  System.out.println("\n MENU \n 1.Insert \n 2.Update \n 3.Delete \n
4.Display \n 5.Exit \n Enter your choice:");
  ch=Integer.parseInt(br.readLine());
  switch(ch)
 {
case 1:
          System.out.println("Enter employee number,name and salary");
          no=Integer.parseInt(br.readLine());
          name=br.readLine();
          sal=Integer.parseInt(br.readLine());
          stmt.executeUpdate("insert into
employee(eno,ename,salary)values("+no+",'"+name+"',"+sal+")");
          System.out.println("Records inserted");
          break;
case 2:
         System.out.println("Enter employee number of the record to be updated:");
         no=Integer.parseInt(br.readLine());
         System.out.println("Enter the new name and salary");
         name=br.readLine();
         sal=Integer.parseInt(br.readLine());
         try
         {
            conn.setAutoCommit(false);
            upc=stmt.executeUpdate("update employee set
ename='"+name+"',salary="+sal+"where eno="+no);
           if(upc!=0)
           {
             conn.commit();
             System.out.println("Records Updated");
           }
           else
          {
            System.out.println("No such records exist");
            break;
           }
         }
          catch(SQLException e)
          {
```

```
            System.out.println("Exception occured:"+e+"\n Records not Updated
\n");
            conn.rollback();

          }
        break;
case 3:
        System.out.println("Enter employee number of the record you want to
delete:");
        no=Integer.parseInt(br.readLine());
        try
        {
          conn.setAutoCommit(false);
          upc=stmt.executeUpdate("delete from employee where eno="+no);
          if(upc!=0)
          {
            System.out.println("Records deleted");
            conn.commit();
          }
          else
          {
             System.out.println("No such records exist");
             break;
           }
         }
        catch(SQLException e)
         {
           System.out.println("Exception occured:"+e+" \n Records not
deleted\n");
            conn.rollback();
         }
        break;
case 4:
        ResultSet rs1=stmt.executeQuery("select * from employee");
        System.out.println("eno \t ename \t salary \n");
        while(rs1.next())
        {

System.out.println(rs1.getInt("eno")+"\t"+rs1.getString("ename")+"\t"+rs1.getInt
("salary"));
```

```
            }
            break;
case 5:
           System.exit(0);

            System.out.println("Enter a valid choice");

}
}
while(ch!=5);
rs.close();
stmt.close();
conn.close();
}
catch(SQLException e)
{
   System.out.println("connection failed:" +e.getMessage());
}
}
}
```

OUTPUT:


Command Prompt - java employee
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.
C:\Users\Gurudev_2>cd C:\Program Files\Java\jdk1.5.0\bin\sreelakshmi3bca
C:\Program Files\Java\jdk1.5.0\bin\sreelakshmi3bca> javac employee.java
C:\Program Files\Java\jdk1.5.0\bin\sreelakshmi3bca>java employee
.....current records.....
ENO ENAME SALARY
1 amrutha 50000
2 neethu 40000
3 nilina 30000
4 lakshmi 20000
5 anju 10000
6 lachu 75000
MENU
1.Insert
2.Update
3.Delete
4.Display
5.Exit
Enter your choice:
1
Enter employee number,name and salary
1
amrutha
10000
Records inserted
MENU
11:25 AM
1/10/2019



PROGRAM -2

JDBC program to connect to student table. implement the record scrolling function-first(),last(),next(),previous(),beforeFirst(),afterLast(),absolute() and relative().

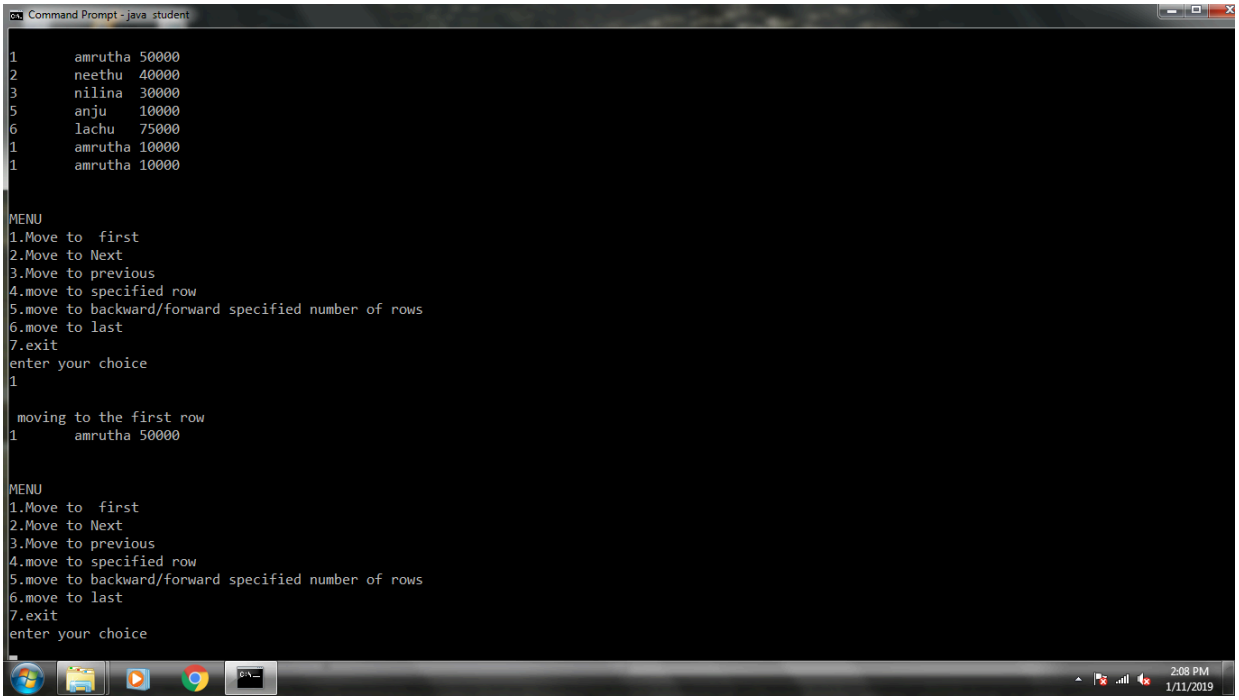
```
import java.sql.*;
import java.io.*;
public class student
  {

    static final String JDBC_DRIVER="com.mysql.jdbc.Driver";
    static final String DB_URL="jdbc:mysql://localhost/abcd";
    static final String USER="root";
    static final String PASS="gurudev";

public static void main(String[]args)throws IOException
     {
int ch,row,rows;
BufferedReader br=new BufferedReader(newInputStreamReader(System.in));
try
{
Class.forName("com.mysql.jdbc.Driver");
```

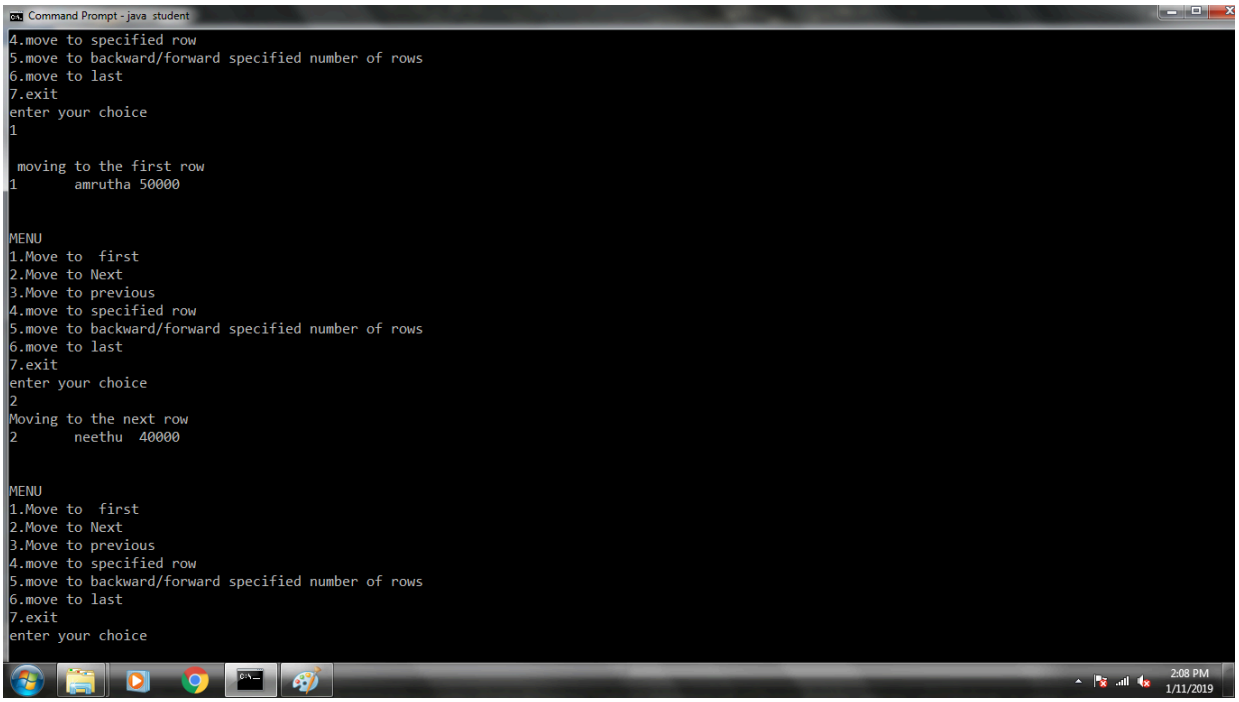
```
}
catch(ClassNotFoundException e)
{
System.out.println("Unable to load the driver");
}

try
{
Connection con=DriverManager.getConnection(DB_URL,USER,PASS);
Statement stmt=con.createStatement();
ResultSet rs=stmt.executeQuery("select * from employee");
System.out.println("\n...CURRENT TABLE...");
System.out.println("eno\t ename \t salary\n");
while(rs.next())
{
System.out.println(rs.getInt("eno")+"\t"+rs.getString("ename")+"\t"+rs.getInt("sa
lary"));
}
do
{
System.out.println("\n\nMENU\n1.Move to  first\n2.Move to Next\n3.Move to
previous\n4.move to specified row\n5.move to backward/forward specified
number of rows\n6.move to last\n7.exit\nenter your choice");
ch=Integer.parseInt(br.readLine());
switch(ch)
{
case 1:
rs.first();
System.out.println("\n moving to the first row");
System.out.println(rs.getInt("eno")+"\t"+rs.getString("ename")+"\t"+rs.getInt("sa
lary"));
break;
case 2:
rs.next();
System.out.println("Moving to the next row");
System.out.println(rs.getInt("eno")+"\t"+rs.getString("ename")+"\t"+rs.getInt("sa
lary"));
break;
```

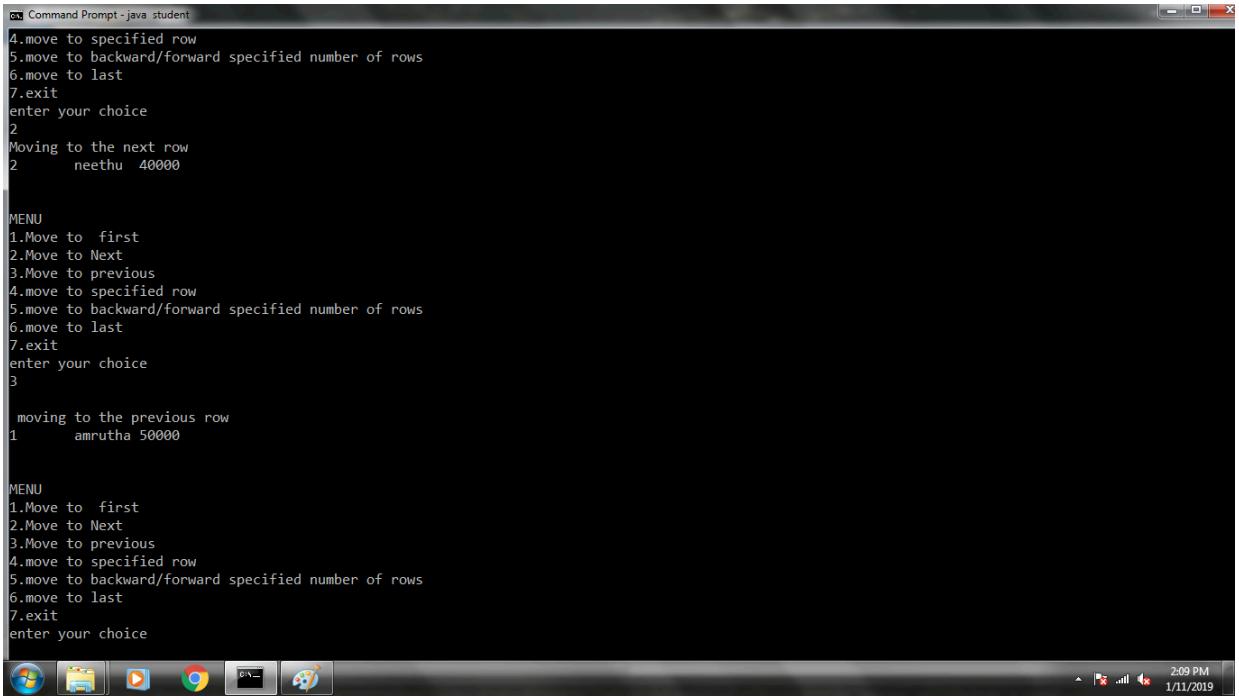
```
case 3:rs.previous();
System.out.println("\n moving to the previous row");
System.out.println(rs.getInt("eno")+"\t"+rs.getString("ename")+"\t"+rs.getInt("sa
lary"));
break;


case 4:
System.out.println("Enter the row  no :");
row=Integer.parseInt(br.readLine());
rs.absolute(row);
System.out.println("\n moving to the specified row");
System.out.println(rs.getInt("eno")+"\t"+rs.getString("ename")+"\t"+rs.getInt("sa
lary"));
break;


case 5:
System.out.println("Enter the  no of rows you want to be moved
forward/backward:");
rows=Integer.parseInt(br.readLine());
rs.relative(rows);
System.out.println("\n moving to the specified row");
break;


case 6:
rs.last();
System.out.println("\n moving to the last row");
System.out.println(rs.getInt("eno")+"\t"+rs.getString("ename")+"\t"+rs.getInt("sa
lary"));
break;


case 7:
System.exit(0);
}
}
 while(ch!=7);
```

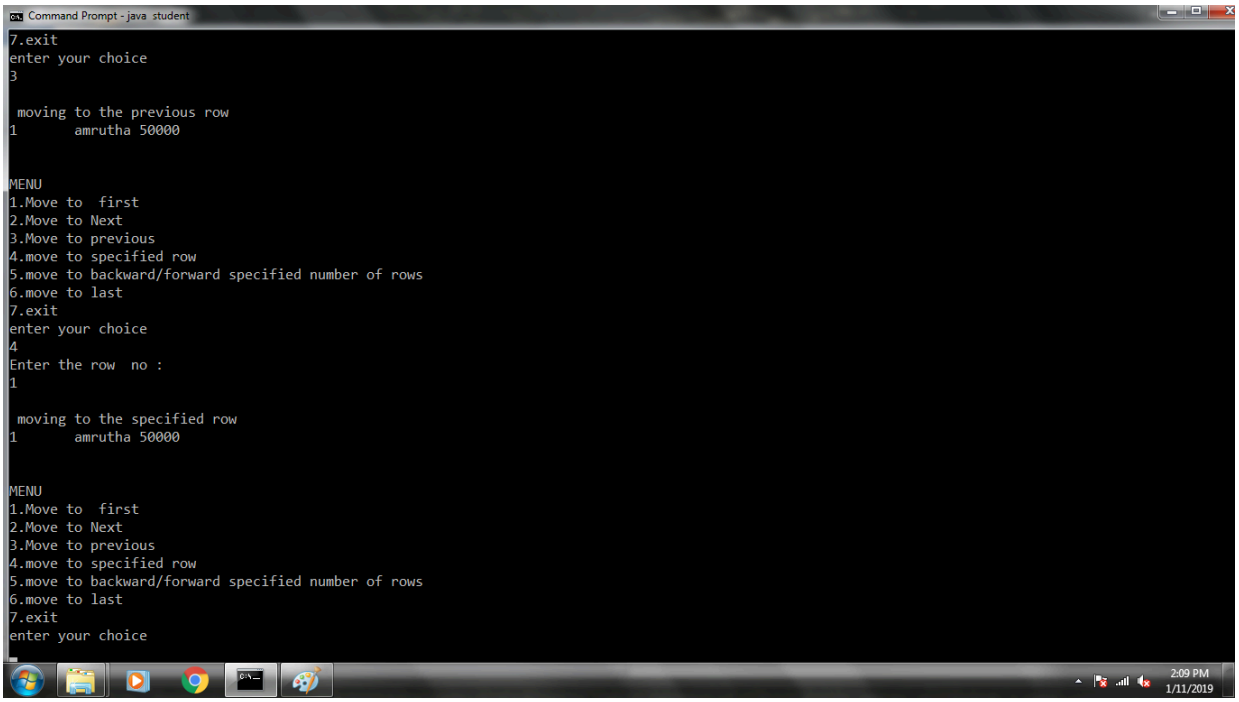
```
rs.close();
stmt.close();
con.close();
}
catch(SQLException e)
{
System.out.println("Connection failed:"+e.getMessage());
e.printStackTrace(System.out);
}
}
}
```

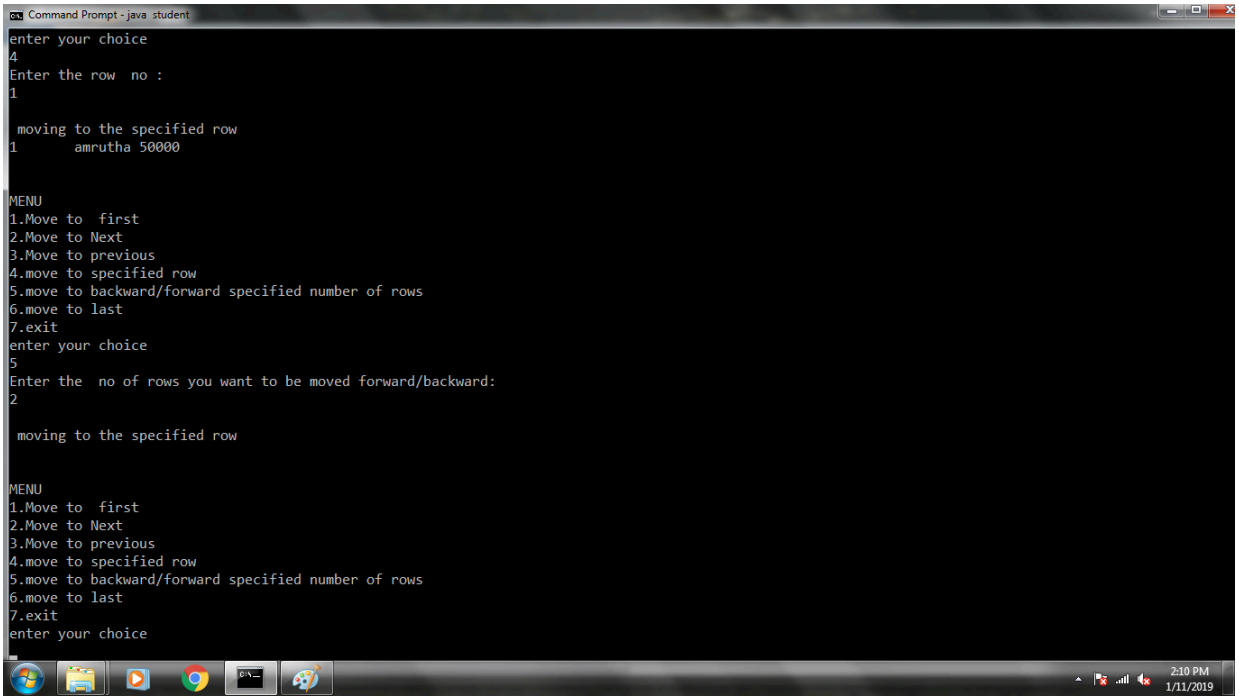
OUTPUT:

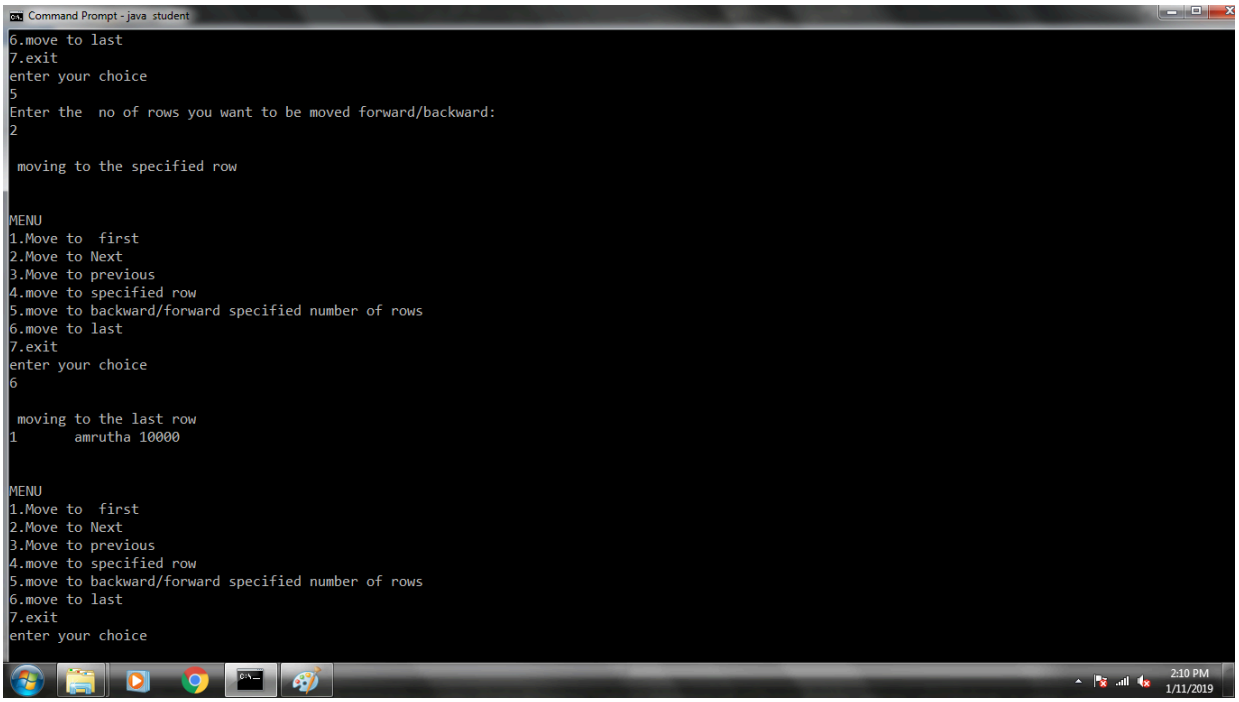

Command Prompt - java student
1 amrutha 50000
2 neethu 40000
3 nilina 30000
5 anju 10000
6 lachu 75000
1 amrutha 10000
1 amrutha 10000
MENU
1.Move to first
2.Move to Next
3.Move to previous
4.move to specified row
5.move to backward/forward specified number of rows
6.move to last
7.exit
enter your choice
1
moving to the first row
1 amrutha 50000
MENU
1.Move to first
2.Move to Next
3.Move to previous
4.move to specified row
5.move to backward/forward specified number of rows
6.move to last
7.exit
enter your choice
2:08 PM
1/11/2019


Command Prompt - java student
4.move to specified row
5.move to backward/forward specified number of rows
6.move to last
7.exit
enter your choice
1
moving to the first row
1 amrutha 50000
MENU
1.Move to first
2.Move to Next
3.Move to previous
4.move to specified row
5.move to backward/forward specified number of rows
6.move to last
7.exit
enter your choice
2
Moving to the next row
2 neethu 40000
MENU
1.Move to first
2.Move to Next
3.Move to previous
4.move to specified row
5.move to backward/forward specified number of rows
6.move to last
7.exit
enter your choice
2:08 PM
1/11/2019


Command Prompt - java student
4.move to specified row
5.move to backward/forward specified number of rows
6.move to last
7.exit
enter your choice
2
Moving to the next row
2 neethu 40000
MENU
1.Move to first
2.Move to Next
3.Move to previous
4.move to specified row
5.move to backward/forward specified number of rows
6.move to last
7.exit
enter your choice
3
moving to the previous row
1 amrutha 50000
MENU
1.Move to first
2.Move to Next
3.Move to previous
4.move to specified row
5.move to backward/forward specified number of rows
6.move to last
7.exit
enter your choice
2:09 PM
1/11/2019


Command Prompt - java student
7.exit
enter your choice
3
moving to the previous row
1 amrutha 50000
MENU
1.Move to first
2.Move to Next
3.Move to previous
4.move to specified row
5.move to backward/forward specified number of rows
6.move to last
7.exit
enter your choice
4
Enter the row no :
1
moving to the specified row
1 amrutha 50000
MENU
1.Move to first
2.Move to Next
3.Move to previous
4.move to specified row
5.move to backward/forward specified number of rows
6.move to last
7.exit
enter your choice
2:09 PM
1/11/2019


Command Prompt - java student
enter your choice
4
Enter the row no :
1
moving to the specified row
1 amrutha 50000
MENU
1.Move to first
2.Move to Next
3.Move to previous
4.move to specified row
5.move to backward/forward specified number of rows
6.move to last
7.exit
enter your choice
5
Enter the no of rows you want to be moved forward/backward:
2
moving to the specified row
MENU
1.Move to first
2.Move to Next
3.Move to previous
4.move to specified row
5.move to backward/forward specified number of rows
6.move to last
7.exit
enter your choice
2:10 PM
1/11/2019


Command Prompt - java student
6.move to last
7.exit
enter your choice
5
Enter the no of rows you want to be moved forward/backward:
2
moving to the specified row
MENU
1.Move to first
2.Move to Next
3.Move to previous
4.move to specified row
5.move to backward/forward specified number of rows
6.move to last
7.exit
enter your choice
6
moving to the last row
1 amrutha 10000
MENU
1.Move to first
2.Move to Next
3.Move to previous
4.move to specified row
5.move to backward/forward specified number of rows
6.move to last
7.exit
enter your choice
2:10 PM
1/11/2019

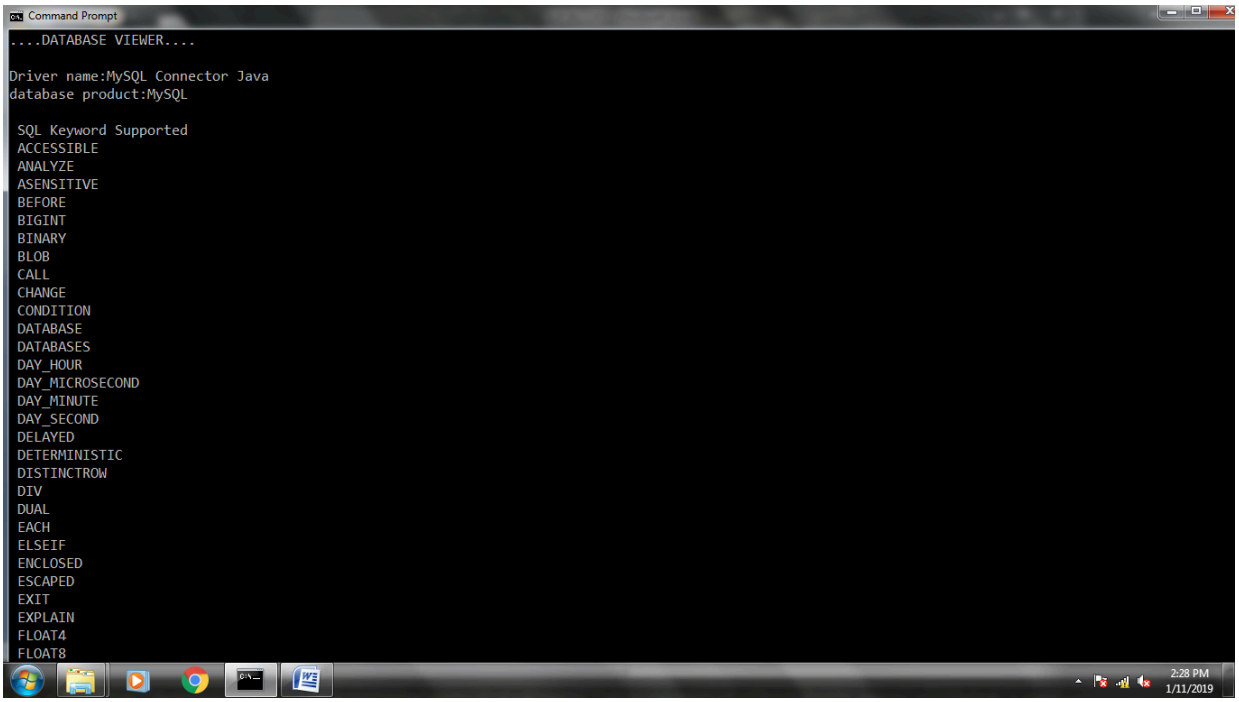
PROGRAM-3

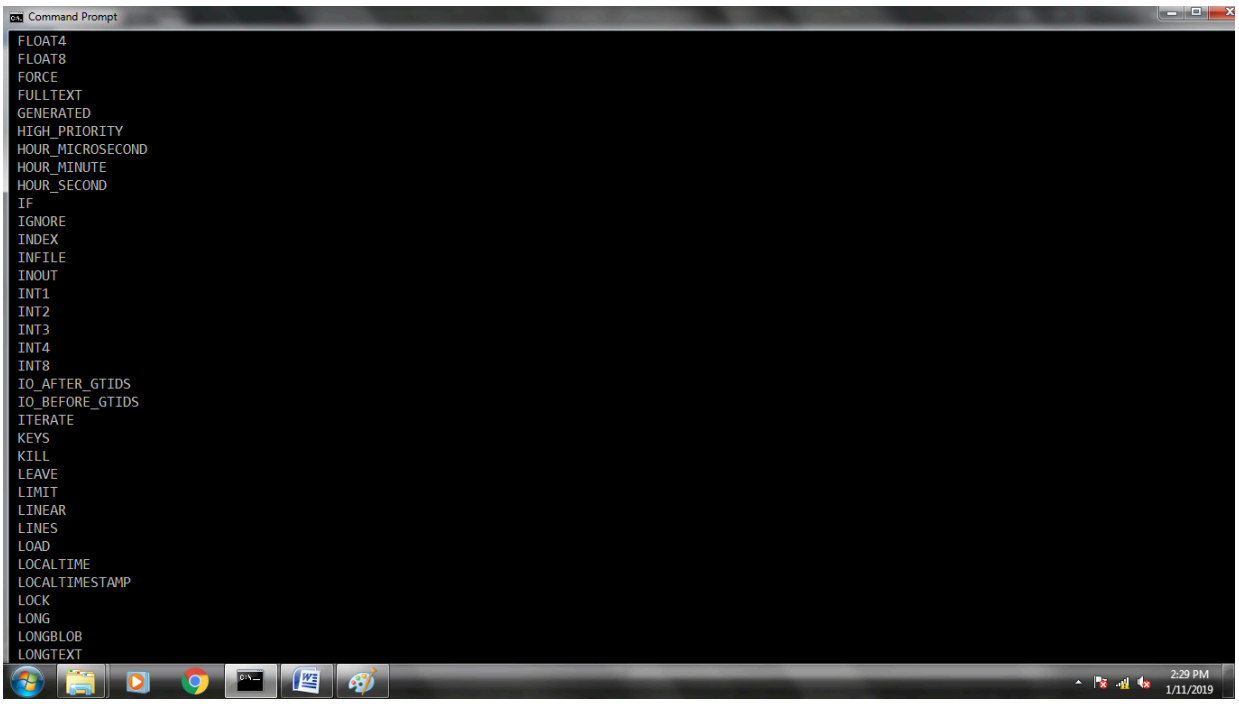
JDBC to display database metadata.

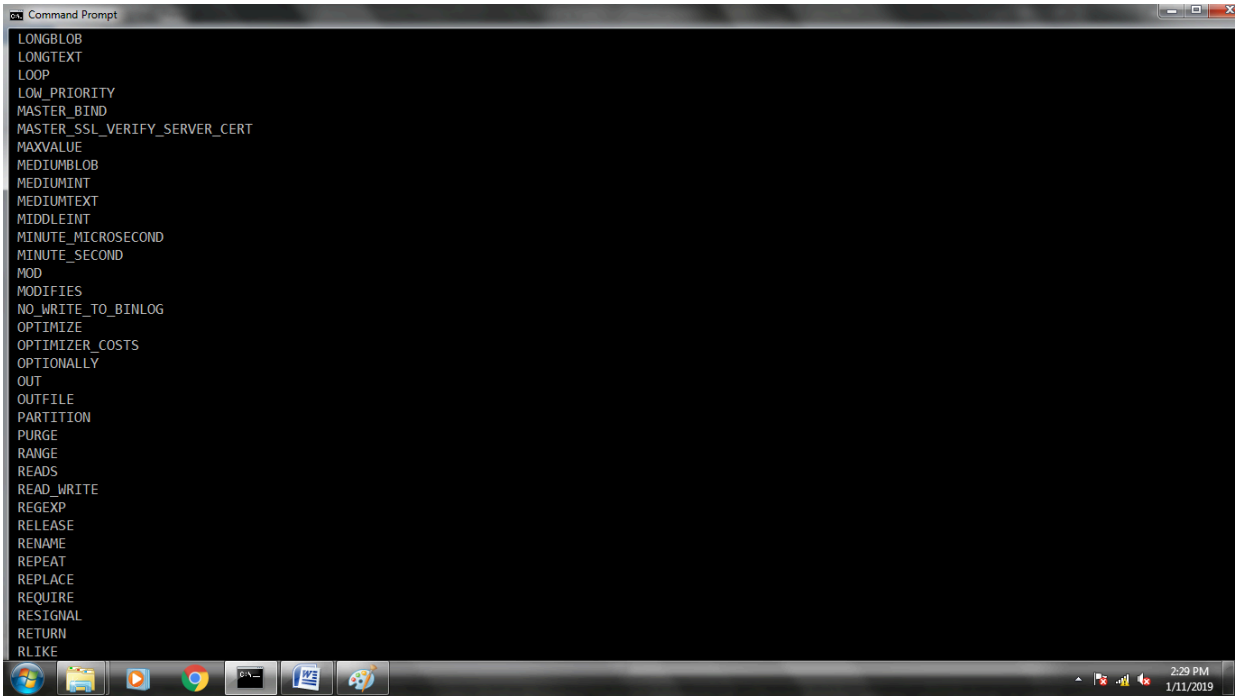
```
import java.sql.*;
import java.io.*;
import java.util.StringTokenizer;
public class DBViewer
{
static final String JDBC_DRIVER="com.mysql.jdbc.Driver";
static final String DB_URL="jdbc:mysql://localhost/abcd";
static final String USER="root";
static final String PASS="gurudev";
public static void main(String args[])
{
System.out.println("\n....DATABASE VIEWER....\n");
{
try
{
Class.forName(JDBC_DRIVER);
Connection con=DriverManager.getConnection(DB_URL,USER,PASS);
DatabaseMetaData dbmd=con.getMetaData();
System.out.println("Driver name:"+dbmd.getDriverName());
System.out.println("database product:"+dbmd.getDatabaseProductName());
System.out.println("\n SQL Keyword Supported");
StringTokenizer st=new StringTokenizer(dbmd.getSQLKeywords(),",");
while(st.hasMoreTokens())
System.out.println(" "+st.nextToken());
String[]tableTypes={"TABLES"};
ResultSet rs=dbmd.getTables(null,null,null,tableTypes);
while(rs.next())
{
String table_name=rs.getString("TABLE_TYPE");
System.out.println("indexes");
ResultSet ilist=dbmd.getIndexInfo(null,null,table_name,false,false);
while(ilist.next())
{
System.out.println("indexname:"+ilist.getString("INDEX_NAME"));
System.out.println("coloumnname:"+ilist.getString("COLOUMN NAME"));
}
```

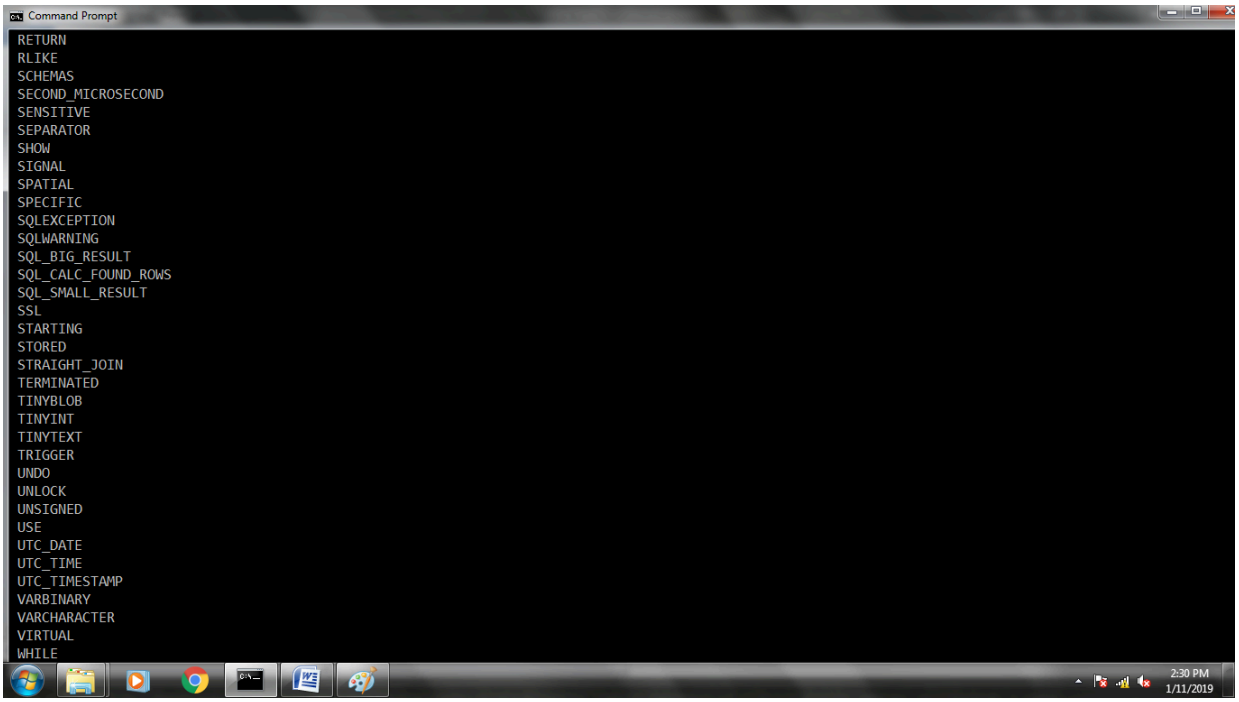
```
ilist.close();
}
rs.close();
con.close();
}
catch(ClassNotFoundException e)
{
System.out.println("unable to load driver");
}
catch(SQLException e)
{
System.out.println("Exception:"+e);
}
}
}
}
```

OUTPUT:


Command Prompt
....DATABASE VIEWER....
Driver name:MySQL Connector Java
database product:MySQL
SQL Keyword Supported
ACCESSIBLE
ANALYZE
ASENSITIVE
BEFORE
BIGINT
BINARY
BLOB
CALL
CHANGE
CONDITION
DATABASE
DATABASES
DAY_HOUR
DAY_MICROSECOND
DAY_MINUTE
DAY_SECOND
DELAYED
DETERMINISTIC
DISTINCTROW
DIV
DUAL
EACH
ELSEIF
ENCLOSED
ESCAPED
EXIT
EXPLAIN
FLOAT4
FLOAT8
2:28 PM
1/11/2019


Command Prompt
FLOAT4
FLOAT8
FORCE
FULLTEXT
GENERATED
HIGH_PRIORITY
HOUR_MICROSECOND
HOUR_MINUTE
HOUR_SECOND
IF
IGNORE
INDEX
INFILE
INOUT
INT1
INT2
INT3
INT4
INT8
IO_AFTER_GTIDS
IO_BEFORE_GTIDS
ITERATE
KEYS
KILL
LEAVE
LIMIT
LINEAR
LINES
LOAD
LOCALTIME
LOCALTIMESTAMP
LOCK
LONG
LONGBLOB
LONGTEXT
2:29 PM
1/11/2019


Command Prompt
LONGBLOB
LONGTEXT
LOOP
LOW_PRIORITY
MASTER_BIND
MASTER_SSL_VERIFY_SERVER_CERT
MAXVALUE
MEDIUMBLOB
MEDIUMINT
MEDIUMTEXT
MIDDLEINT
MINUTE_MICROSECOND
MINUTE_SECOND
MOD
MODIFIES
NO_WRITE_TO_BINLOG
OPTIMIZE
OPTIMIZER_COSTS
OPTIONALLY
OUT
OUTFILE
PARTITION
PURGE
RANGE
READS
READ_WRITE
REGEXP
RELEASE
RENAME
REPEAT
REPLACE
REQUIRE
RESIGNAL
RETURN
RLIKE
2:29 PM
1/11/2019

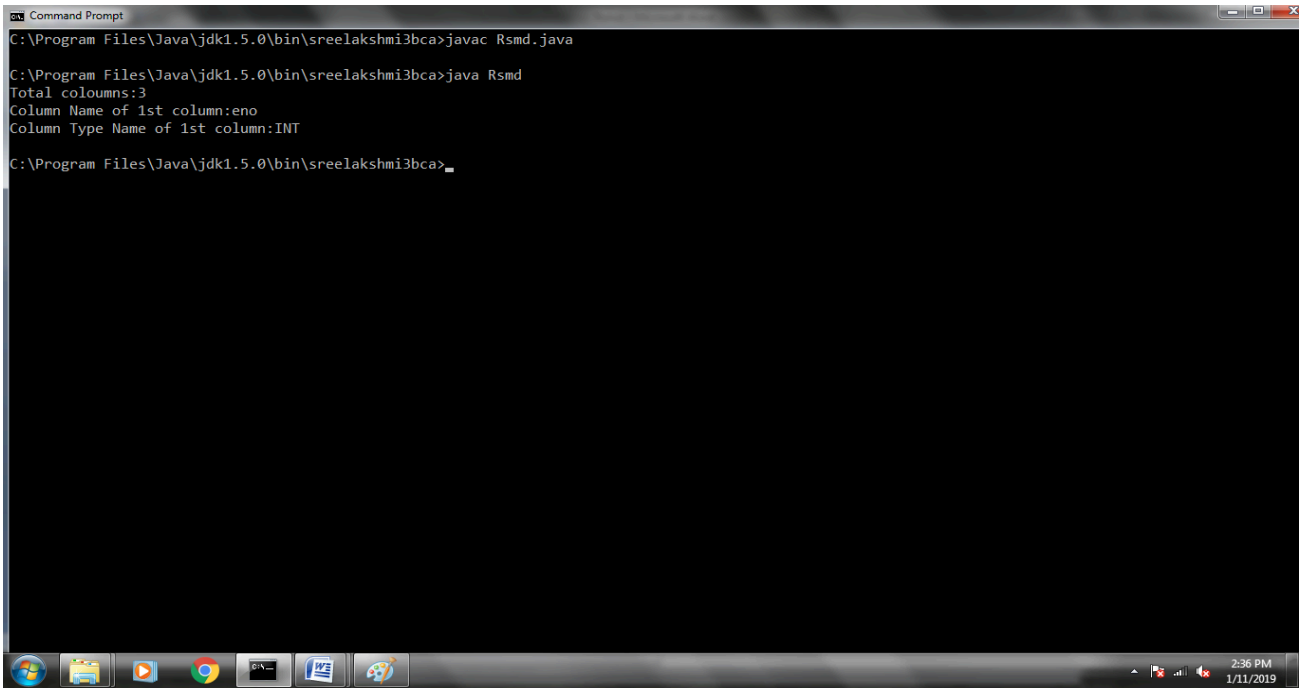

Command Prompt
RETURN
RLIKE
SCHEMAS
SECOND_MICROSECOND
SENSITIVE
SEPARATOR
SHOW
SIGNAL
SPATIAL
SPECIFIC
SQLEXCEPTION
SQLWARNING
SQL_BIG_RESULT
SQL_CALC_FOUND_ROWS
SQL_SMALL_RESULT
SSL
STARTING
STORED
STRAIGHT_JOIN
TERMINATED
TINYBLOB
TINYINT
TINYTEXT
TRIGGER
UNDO
UNLOCK
UNSIGNED
USE
UTC_DATE
UTC_TIME
UTC_TIMESTAMP
VARBINARY
VARCHARACTER
VIRTUAL
WHILE
2:30 PM
1/11/2019

PROGRAM -4

JDBC program to display Resultset metadata.

```
import java.sql.*;
class Rsmd
{
	static final String JDBC_DRIVER="com.mysql.jdbc.Driver";
	static final String DB_URL="jdbc:mysql://localhost/abcd";
	static final String USER="root";
static final String PASS="gurudev";
static final String table="emp1";
public static void main(String args[])
{
	try
	{
		Class.forName("com.mysql.jdbc.Driver");
		Connection
con=DriverManager.getConnection(DB_URL,USER,PASS);
		PreparedStatement ps=con.prepareStatement("select * from emp1");
ResultSet rs=ps.executeQuery();
ResultSetMetaData rsmd=rs.getMetaData();
System.out.println("Total coloumns:"+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:


Command Prompt
C:\Program Files\Java\jdk1.5.0\bin\sreelakshmi3bca>javac Rsmd.java
C:\Program Files\Java\jdk1.5.0\bin\sreelakshmi3bca>java Rsmd
Total coloumns:3
Column Name of 1st column:eno
Column Type Name of 1st column:INT
C:\Program Files\Java\jdk1.5.0\bin\sreelakshmi3bca>
2:36 PM
1/11/2019

PROGRAM -5

RMI program for complex number operation.

Complex Client.java

```
import java.rmi.Naming;
import java .io.*;
public class ComplexClient
{
public static void main(String args[])throws IOException
{
BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
int x,y,z,w,ans1,ans2;
System.out.println("enter 1st complex numbers:");
x=Integer.parseInt(br.readLine());
y=Integer.parseInt(br.readLine());
System.out.println("enter @nd complex number:");
z=Integer.parseInt(br.readLine());
w=Integer.parseInt(br.readLine());
try
{
ComplexInter obj=(ComplexInter)Naming.lookup("rmi://localhost/com");
ans1=obj.add1(x,z);
ans2=obj.add2(y,w);
System.out.println("sum="+ans1+"+i"+ans2);
}
catch(Exception e)
{
System.out.println("Error:"+e);
}
}
}
```

ComplexImpl.java

```
import java.rmi.server.UnicastRemoteObject;
import java.rmi.RemoteException;
public class ComplexImpl extends UnicastRemoteObject implements
ComplexInter
{
public ComplexImpl()throws RemoteException{}
```

```
public int add1(int a1,int b1)throws RemoteException
{
int c=a1+b1;
return(c);
}
public int add2(int a2,int b2)throws RemoteException
{
int d=a2+b2;
return(d);
}
}
```

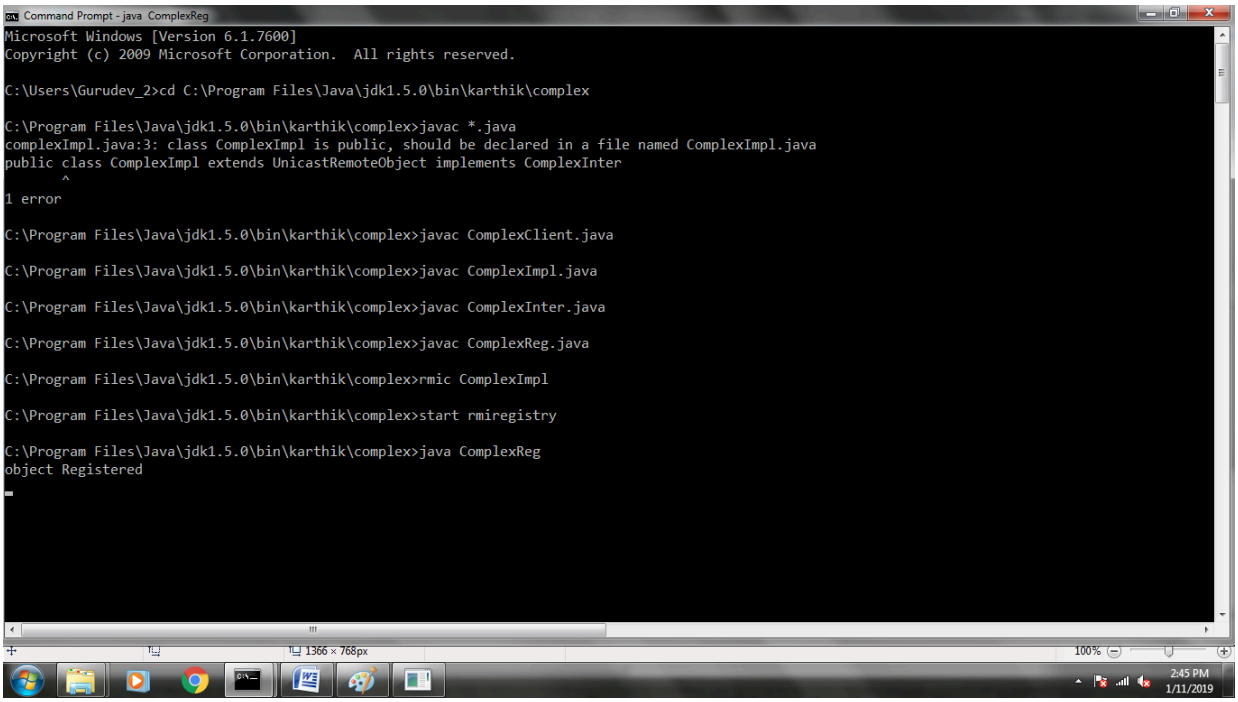
complexInter.java

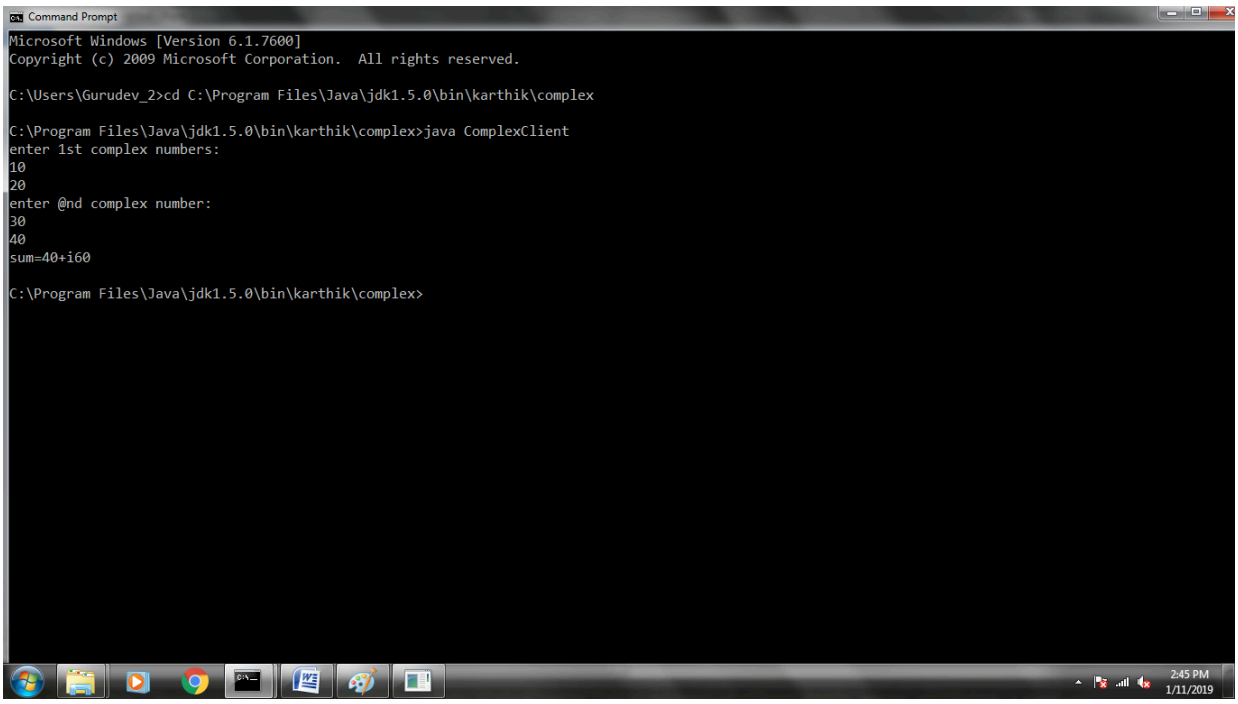
```
import java.rmi.Remote;
import java.rmi.RemoteException;
public interface ComplexInter extends java.rmi.Remote
{
public int add1(int a1,int b1)throws RemoteException;
public int add2(int a2,int b2)throws RemoteException;
}
```

ComplexReg.java

```
import java.rmi.Naming;
public class ComplexReg
{
public static void main(String artgs[])
{
try
{
ComplexImpl comp=new ComplexImpl();
Naming.rebind("com",comp);
System.out.println("object Registered");
}
catch(Exception e)
{
e.printStackTrace();
}
}
}
```

OUTPUT:


Command Prompt - java ComplexReg
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.
C:\Users\Gurudev_2>cd C:\Program Files\Java\jdk1.5.0\bin\karthik\complex
C:\Program Files\Java\jdk1.5.0\bin\karthik\complex>javac *.java
complexImpl.java:3: class ComplexImpl is public, should be declared in a file named ComplexImpl.java
public class ComplexImpl extends UnicastRemoteObject implements ComplexInter
^
1 error
C:\Program Files\Java\jdk1.5.0\bin\karthik\complex>javac ComplexClient.java
C:\Program Files\Java\jdk1.5.0\bin\karthik\complex>javac ComplexImpl.java
C:\Program Files\Java\jdk1.5.0\bin\karthik\complex>javac ComplexInter.java
C:\Program Files\Java\jdk1.5.0\bin\karthik\complex>javac ComplexReg.java
C:\Program Files\Java\jdk1.5.0\bin\karthik\complex>rmic ComplexImpl
C:\Program Files\Java\jdk1.5.0\bin\karthik\complex>start rmiregistry
C:\Program Files\Java\jdk1.5.0\bin\karthik\complex>java ComplexReg
object Registered
1366 × 768px
100%
2:45 PM
1/11/2019


Command Prompt
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.
C:\Users\Gurudev_2>cd C:\Program Files\Java\jdk1.5.0\bin\karthik\complex
C:\Program Files\Java\jdk1.5.0\bin\karthik\complex>java ComplexClient
enter 1st complex numbers:
10
20
enter @nd complex number:
30
40
sum=40+i60
C:\Program Files\Java\jdk1.5.0\bin\karthik\complex>
2:45 PM
1/11/2019

PROGRAM -6

RMI program for bank operation.

Bank.java

```
import java.rmi.*;
public interface bank extends Remote
{
public float deposit(float accno,float amt)throws RemoteException;
public float withdraw (float accno,float amt)throws RemoteException;
public float balance (float accno)throws RemoteException;
};
```

Bankimpl.java

```
import java.rmi.*;
import java.rmi.server.*;
public class bankimpl extends UnicastRemoteObject implements bank
{
float c=0;
public bankimpl()throws RemoteException
{
super();
}
public float deposit(float accno,float amt)throws RemoteException
{
c=c+amt;
return c;
}
public float withdraw(float accno,float amt)throws RemoteException
{
c=c-amt;
return c;
}
public float balance(float accno)throws RemoteException
{
return c;
}
}
```

Client.java

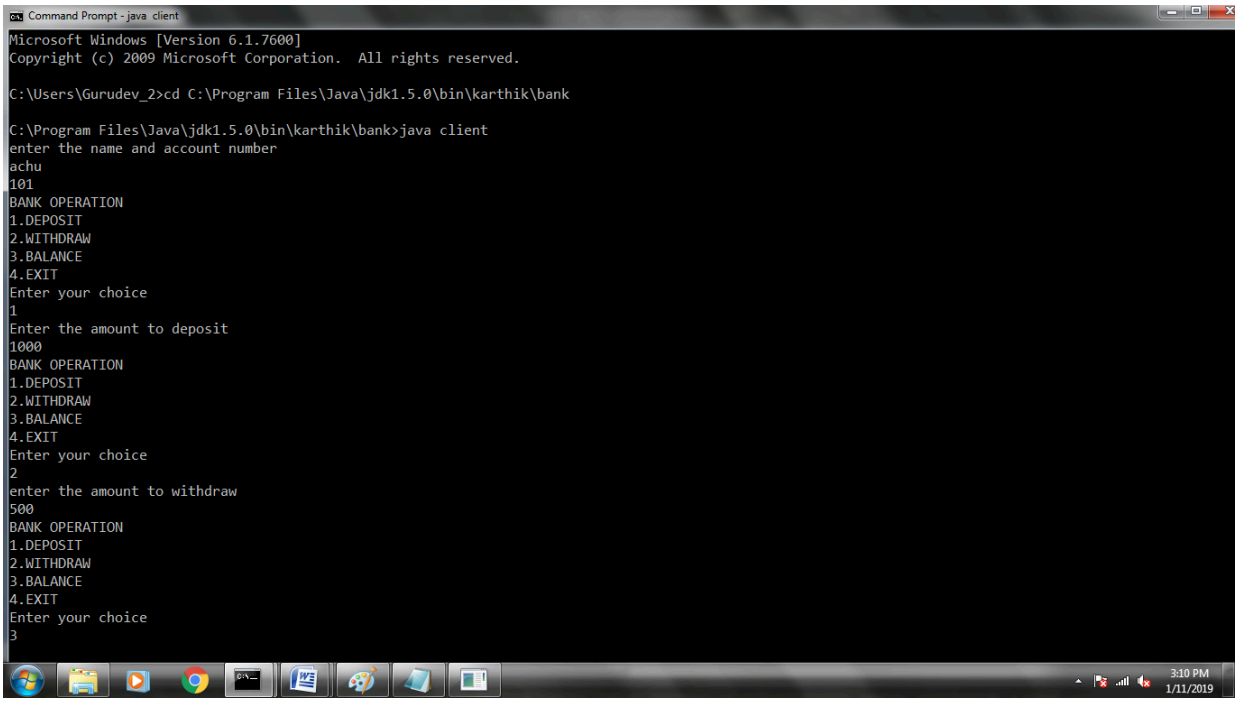
```
import java.rmi.*;
import java.io.*;
public class client
{
public static void main(String args[])
{
float c,act,b,amt;
String name;
int ch;
try
{
BufferedReader rd=new BufferedReader(new InputStreamReader(System.in));
bank ar=(bank)Naming.lookup("bank");
System.out.println("enter the name and account number");
name=rd.readLine();
act=Float.parseFloat(rd.readLine());
do
{
        System.out.println("BANK OPERATION");
        System.out.println("1.DEPOSIT");
        System.out.println("2.WITHDRAW");
        System.out.println("3.BALANCE");
        System.out.println("4.EXIT");
        System.out.println("Enter your choice");
        ch=Integer.parseInt(rd.readLine());
        switch(ch)
        {
                case 1:
                System.out.println("Enter the amount to deposit");
                amt=Float.parseFloat(rd.readLine());
                c=ar.deposit(act,amt);
                break;
                case 2:
                System.out.println("enter the amount to withdraw");
                amt=Float.parseFloat(rd.readLine());
                c=ar.withdraw(act,amt);
                break;
                case 3:
                System.out.println("");
```

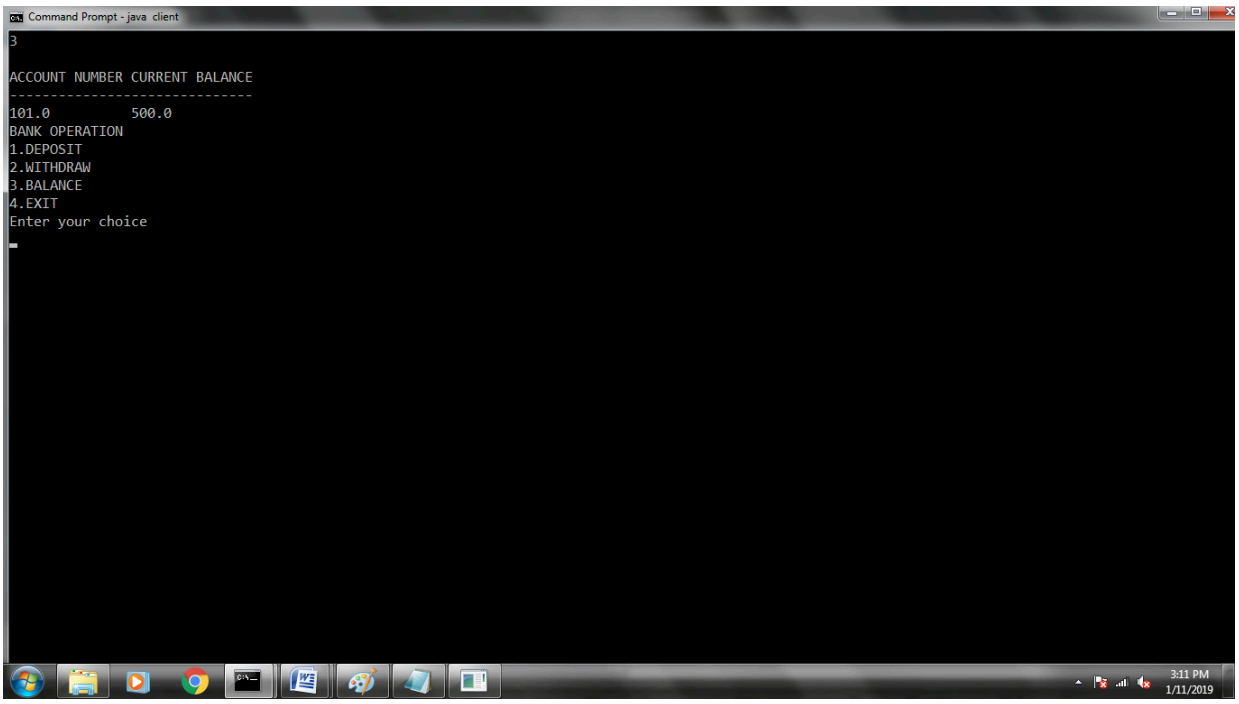
```
                c=ar.balance(act);
                System.out.println("ACCOUNT NUMBER CURRENT
BALANCE");
                System.out.println("------------------------------");
                System.out.println(act+"          "+c);
                break;

        }

} while(ch<4);
}
catch(Exception e)
{

}
}
}
```

Reg.java

```
import java.rmi.*;
public class reg
{
public static void main(String args[])
{
try
{
bankimpl ob=new bankimpl();
Naming.rebind("bank",ob);
}
catch(Exception e)
{
}
}
}
```

OUTPUT:


Command Prompt - java client
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.
C:\Users\Gurudev_2>cd C:\Program Files\Java\jdk1.5.0\bin\karthik\bank
C:\Program Files\Java\jdk1.5.0\bin\karthik\bank>java client
enter the name and account number
achu
101
BANK OPERATION
1.DEPOSIT
2.WITHDRAW
3.BALANCE
4.EXIT
Enter your choice
1
Enter the amount to deposit
1000
BANK OPERATION
1.DEPOSIT
2.WITHDRAW
3.BALANCE
4.EXIT
Enter your choice
2
enter the amount to withdraw
500
BANK OPERATION
1.DEPOSIT
2.WITHDRAW
3.BALANCE
4.EXIT
Enter your choice
3
3:10 PM
1/11/2019


Command Prompt - java client
3
ACCOUNT NUMBER CURRENT BALANCE
101.0 500.0
BANK OPERATION
1.DEPOSIT
2.WITHDRAW
3.BALANCE
4.EXIT
Enter your choice
3:11 PM
1/11/2019

PROGRAM -7

RMI program for matrix addition and subtraction.

Client.java

```
import java.rmi.*;
import java.io.*;
class matt
{
int i,j;
BufferedReader rd=new BufferedReader(new InputStreamReader(System.in));
public int[][]read(int[][] A,int r,int c)throws IOException
{
for(i=0;i<r;i++)
{
for(j=0;j<c;j++)
{
A[i][j]=Integer.parseInt(rd.readLine());
}
}
return A;
}
public void print(int[][]A,int r,int c)throws IOException
{
for(i=0;i<r;i++)
{
for(j=0;j<c;j++)
{
System.out.println(" "+A[i][j]);
}
System.out.println(" ");
}
}
}
public class client
{
public static void main(String args[])
{
int r,c,ch;
matt mt=new matt();
```

```
int[][] A=new int[100][100];
int [][] B=new int[100][100];
int[][] C=new int[100][100];
BufferedReader rd=new BufferedReader(new InputStreamReader(System.in));
try
{
System.out.println("enter the order of matrix");
r=Integer.parseInt(rd.readLine());
c=Integer.parseInt(rd.readLine());
System.out.println("enter the order of first matrix");
A=mt.read(A,r,c);
System.out.println("enter the order of second matrix");
B=mt.read(B,r,c);
System.out.println("FIRST MATRIX");
mt.print(A,r,c);
System.out.println("SECOND MATRIX");
mt.print(B,r,c);
matrix a=(matrix)Naming.lookup("matt");
do
{
System.out.println("");
System.out.println("MATRIX OPERATION");
System.out.println("1.ADDITION");
System.out.println("2.SUBTRACTION");
System.out.println("3.EXIT");
System.out.println("ENTER YOUR CHOICE");
ch=Integer.parseInt(rd.readLine());
switch(ch)
{
case 1:
C=a.add(A,B,r,c);
System.out.println("ADDITION");
mt.print(C,r,c);
break;
case 2:
C=a.sub(A,B,r,c);
System.out.println("SUBTRACTION");
mt.print(C,r,c);
break;
}
```

```
}while(ch<3);
}
catch(Exception e)
{
}
}
}
```

Matrix.java

```
import java.rmi.*;
public interface matrix extends Remote
{
public int[][]add(int[][] A,int[][] B,int r,int c)throws RemoteException;
public int[][]sub(int[][] A,int[][] B,int r,int c)throws RemoteException;
};
```

Matriximpl.java

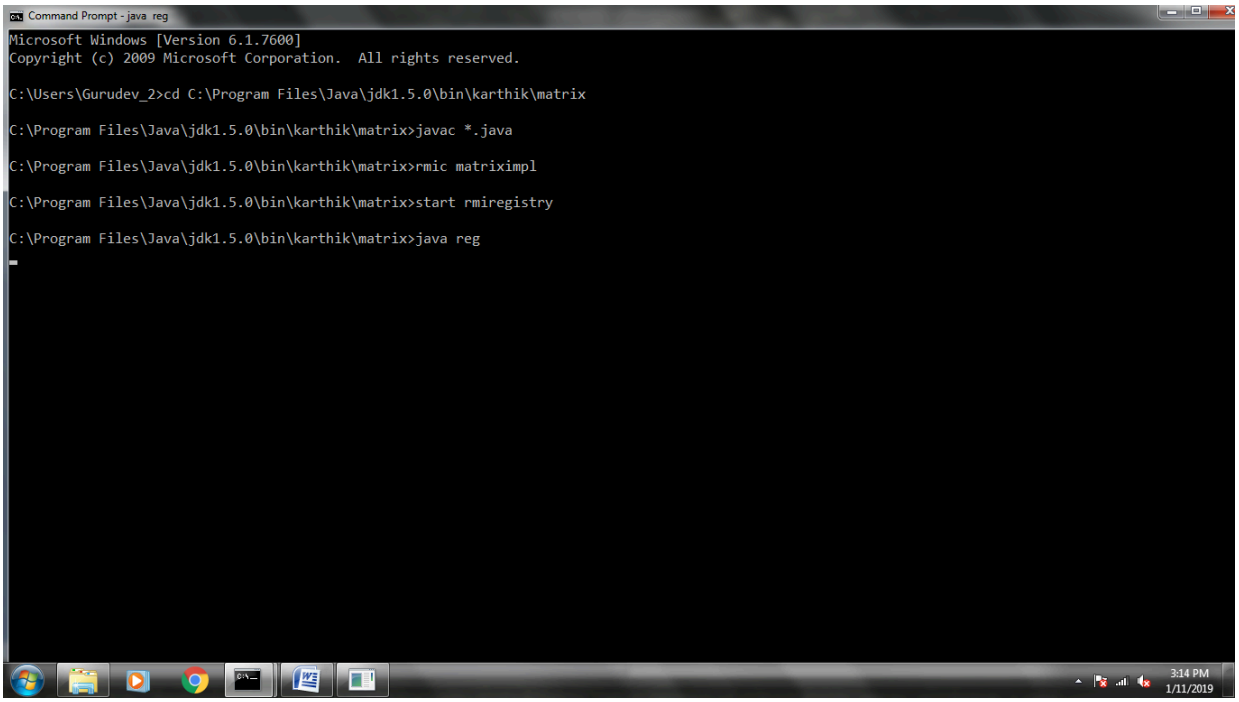
```
import java.rmi.*;
import java.rmi.server.*;
import java.io.*;
public class matriximpl extends UnicastRemoteObject implements matrix
{
String res;
int rsum,i,j;
int[][]C=new int[100][100];
public matriximpl()throws RemoteException
{
super();
}
public int[][]add(int[][] A,int[][] B,int r,int c)throws RemoteException
{
for(i=0;i<r;i++)
{
for(j=0;j<c;j++)
{
C[i][j]=A[i][j]+B[i][j];
}
}
return C;
}
public int[][]sub(int[][] A,int[][] B,int r,int c)throws RemoteException
```

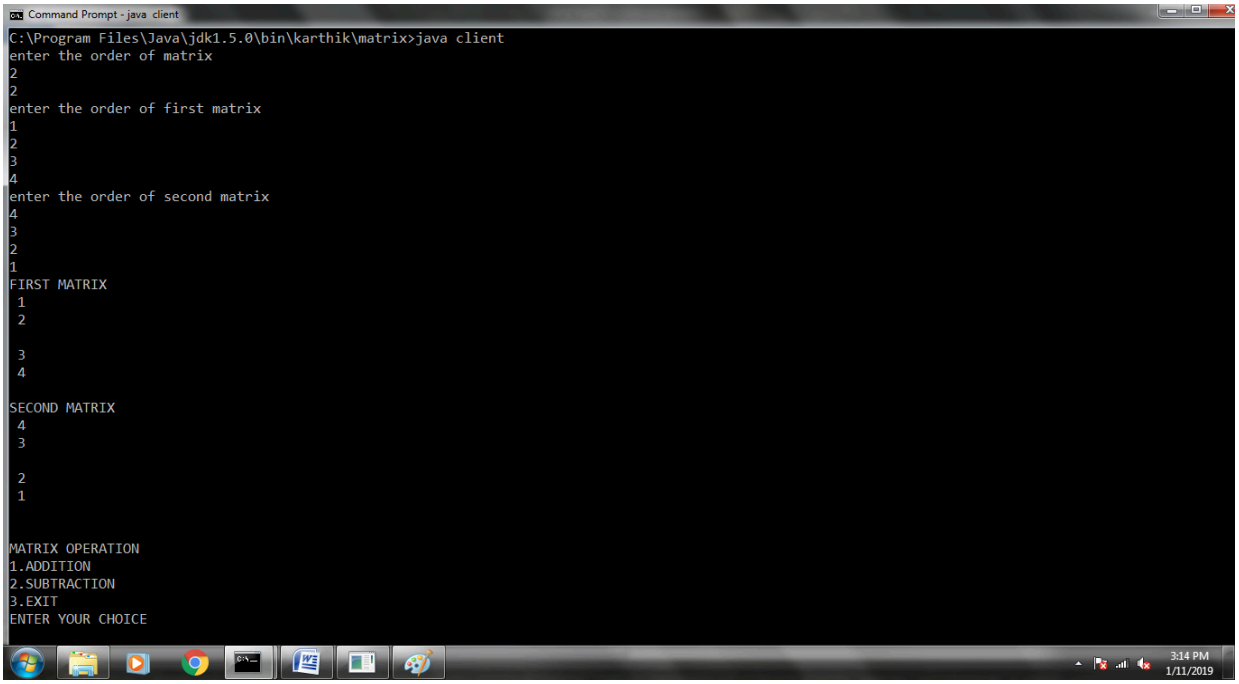
```
{
for(i=0;i<r;i++)
{
for(j=0;j<c;j++)
{
C[i][j]=A[i][j]-B[i][j];
}
}
return C;
}
}
```

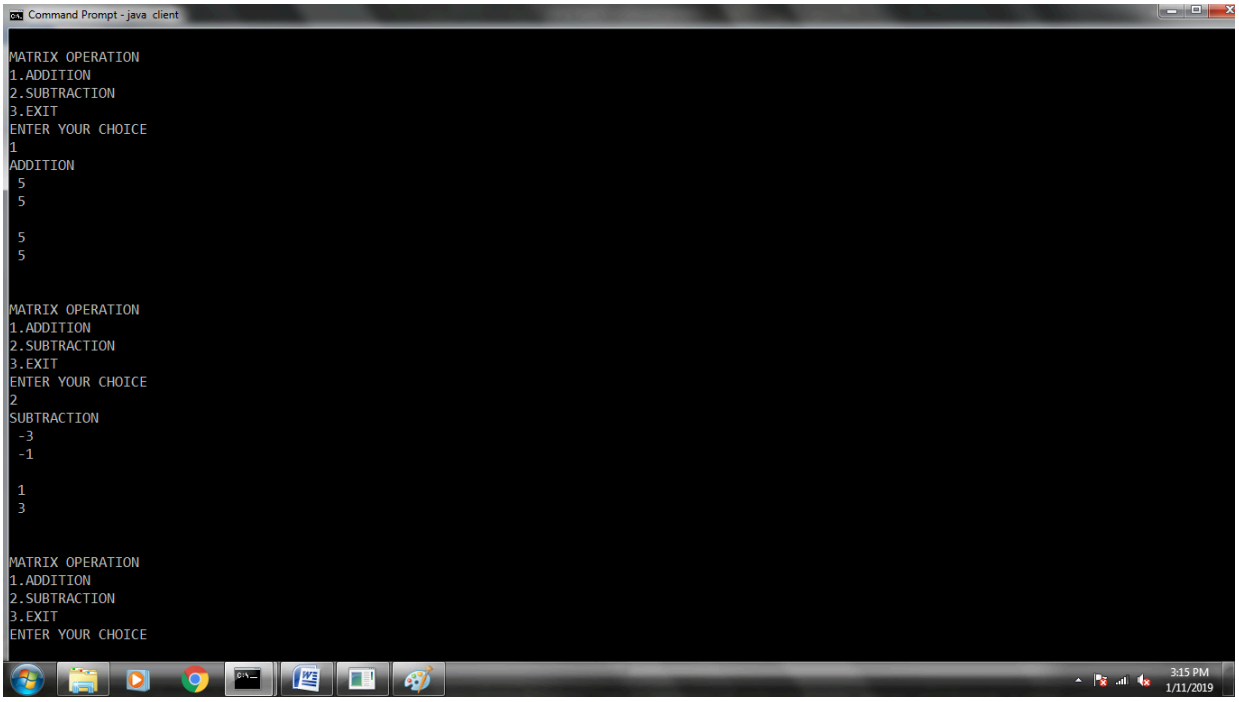
Reg.java

```
import java.rmi.*;
public class reg
{
public static void main(String args[])
{
try
{
matriximpl ob=new matriximpl();
Naming.rebind("matt",ob);
}
catch(Exception e)
{
}
}
}
```

OUTPUT:


Command Prompt - java reg
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.
C:\Users\Gurudev_2>cd C:\Program Files\Java\jdk1.5.0\bin\karthik\matrix
C:\Program Files\Java\jdk1.5.0\bin\karthik\matrix>javac *.java
C:\Program Files\Java\jdk1.5.0\bin\karthik\matrix>rmic matriximpl
C:\Program Files\Java\jdk1.5.0\bin\karthik\matrix>start rmiregistry
C:\Program Files\Java\jdk1.5.0\bin\karthik\matrix>java reg
3:14 PM
1/11/2019


Command Prompt - java client
C:\Program Files\Java\jdk1.5.0\bin\karthik\matrix>java client
enter the order of matrix
2
2
enter the order of first matrix
1
2
3
4
enter the order of second matrix
4
3
2
1
FIRST MATRIX
1
2
3
4
SECOND MATRIX
4
3
2
1
MATRIX OPERATION
1.ADDITION
2.SUBTRACTION
3.EXIT
ENTER YOUR CHOICE
3:14 PM
1/11/2019


Command Prompt - java client
MATRIX OPERATION
1.ADDITION
2.SUBTRACTION
3.EXIT
ENTER YOUR CHOICE
1
ADDITION
5
5
5
5
MATRIX OPERATION
1.ADDITION
2.SUBTRACTION
3.EXIT
ENTER YOUR CHOICE
2
SUBTRACTION
-3
-1
1
3
MATRIX OPERATION
1.ADDITION
2.SUBTRACTION
3.EXIT
ENTER YOUR CHOICE
3:15 PM
1/11/2019

PROGRAM-8

CORBA program for arithmetic operation.

Arithematic.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);
};
```

Arithematicimp.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

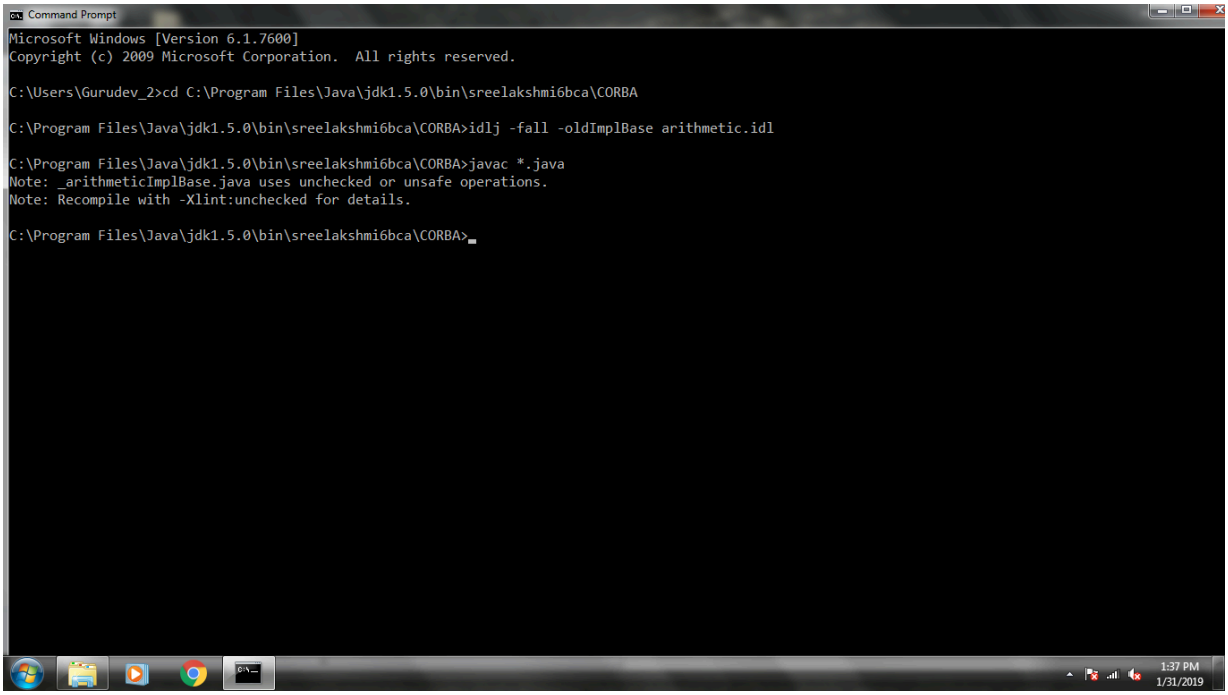
```
import org.omg.CORBA.*;
import org.omg.CosNaming.*;
import java.io.*;
public class client
{
public static void main(String arg[])
{
 try{
        BufferedReader rd=new BufferedReader(new
InputStreamReader(System.in));
       ORB orb=ORB.init(arg,null);
       org.omg.CORBA.Object
ob=orb.resolve_initial_references("NameService");
       NamingContext ctx=NamingContextHelper.narrow(ob);
       NameComponent nc=new NameComponent("Message","");
       NameComponent path[]={nc};
       arithmetic ar=arithmeticHelper.narrow(ctx.resolve(path));
       float c,a,b;
         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);
       }
```

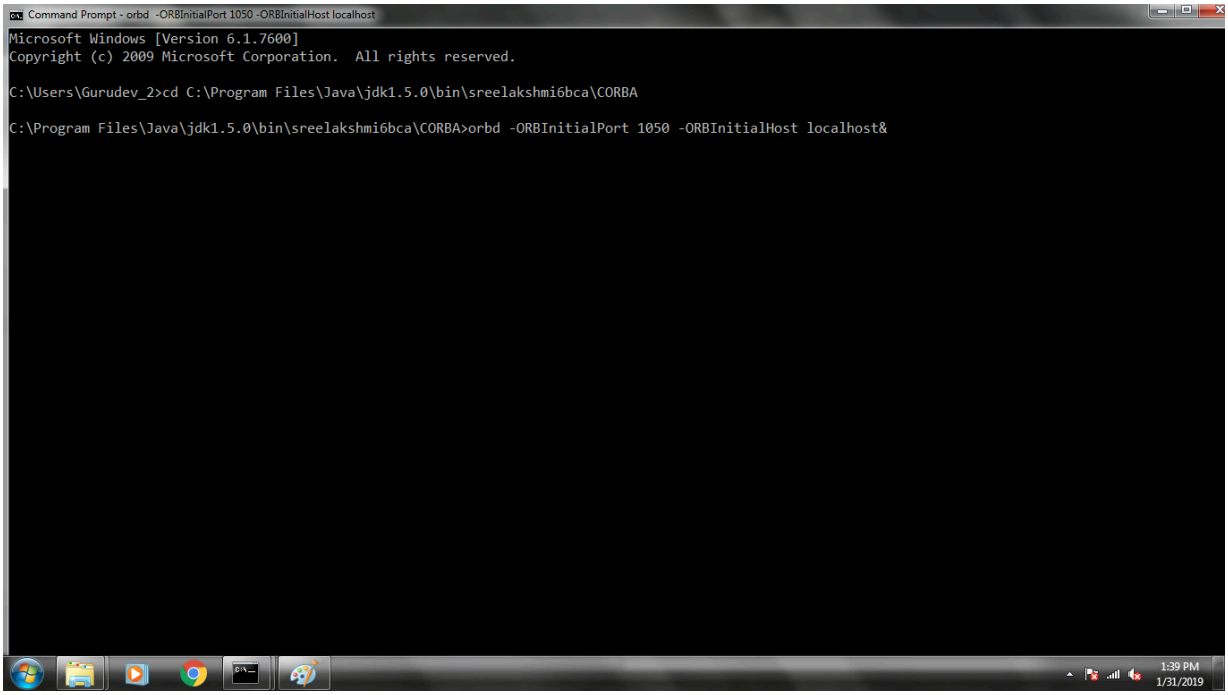
```
        catch(Exception e)
        {
        }
        }
        }
```

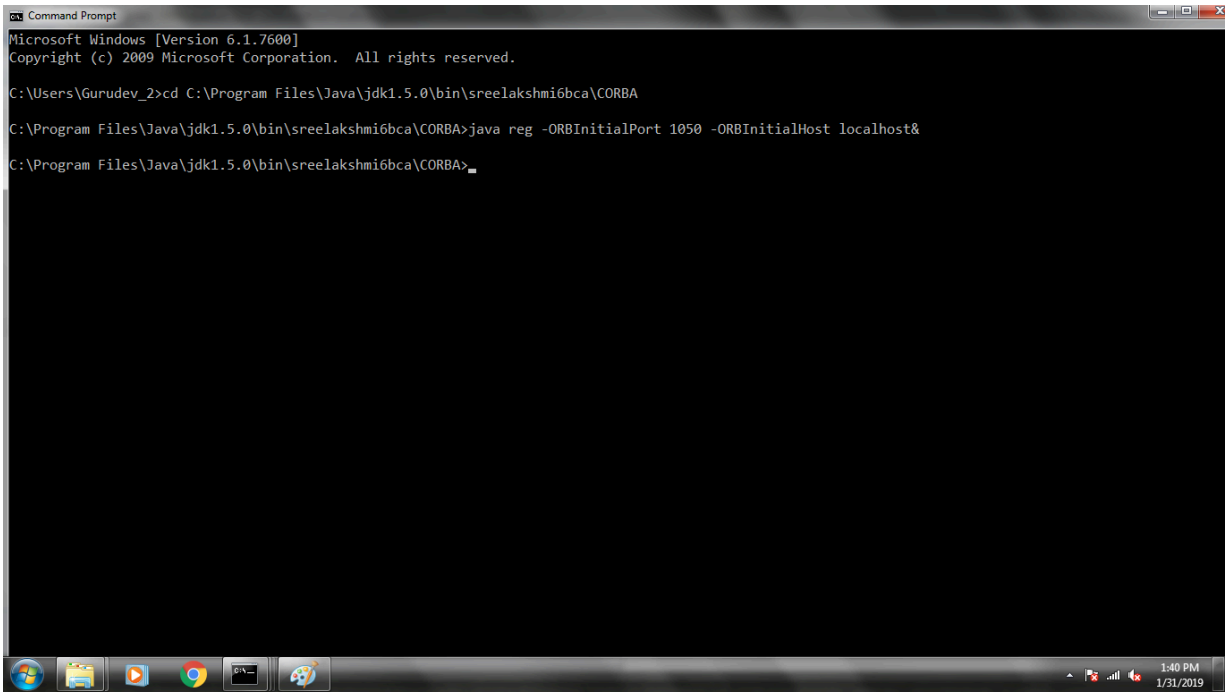
Reg.java

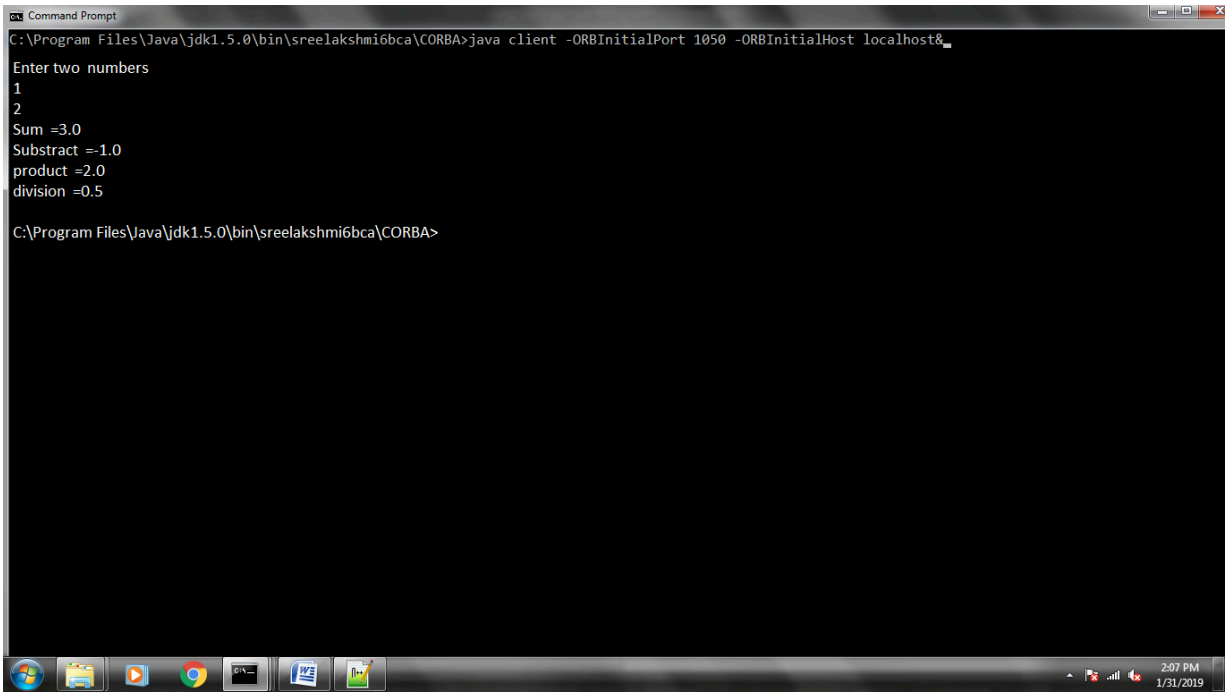
```
import org.omg.CORBA.*;
import org.omg.CosNaming.*;
public class reg
{
public static void main(String arg[])
{
 try{
        ORB orb=ORB.init(arg,null);
        org.omg.CORBA.Object
ob=orb.resolve_initial_references("NameService");
        NamingContext ctx=NamingContextHelper.narrow(ob);
        NameComponent nc=new NameComponent("Message","");
        NameComponent path[]={nc};
        arithmeticimp m=new arithmeticimp();
        ctx.rebind(path,m);
        orb.run();
        }
        catch(Exception e)
        {
        }
        }
        }
```

OUTPUT:


Command Prompt
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.
C:\Users\Gurudev_2>cd C:\Program Files\Java\jdk1.5.0\bin\sreelakshmi6bca\CORBA
C:\Program Files\Java\jdk1.5.0\bin\sreelakshmi6bca\CORBA>idlj -fall -oldImplBase arithmetic.idl
C:\Program Files\Java\jdk1.5.0\bin\sreelakshmi6bca\CORBA>javac *.java
Note: _arithmeticImplBase.java uses unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.
C:\Program Files\Java\jdk1.5.0\bin\sreelakshmi6bca\CORBA>
1:37 PM
1/31/2019


Command Prompt - orbd -ORBInitialPort 1050 -ORBInitialHost localhost
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.
C:\Users\Gurudev_2>cd C:\Program Files\Java\jdk1.5.0\bin\sreelakshmi6bca\CORBA
C:\Program Files\Java\jdk1.5.0\bin\sreelakshmi6bca\CORBA>orbd -ORBInitialPort 1050 -ORBInitialHost localhost&
1:39 PM
1/31/2019


Command Prompt
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.
C:\Users\Gurudev_2>cd C:\Program Files\Java\jdk1.5.0\bin\sreelakshmi6bca\CORBA
C:\Program Files\Java\jdk1.5.0\bin\sreelakshmi6bca\CORBA>java reg -ORBInitialPort 1050 -ORBInitialHost localhost&
C:\Program Files\Java\jdk1.5.0\bin\sreelakshmi6bca\CORBA>
1:40 PM
1/31/2019


Command Prompt
C:\Program Files\Java\jdk1.5.0\bin\sreelakshmi6bca\CORBA>java client -ORBInitialPort 1050 -ORBInitialHost localhost&
Enter two numbers
1
2
Sum =3.0
Substract =-1.0
product =2.0
division =0.5
C:\Program Files\Java\jdk1.5.0\bin\sreelakshmi6bca\CORBA>
2:07 PM
1/31/2019

PROGRAM-9

Create an html form to read student detail such as Roll, name, age, sex, qualification, percentage of marks etc. Write a servlet program that displays the same details.

Hello.java

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class helloform extends HttpServlet
{
protected void doGet(HttpServletRequest req,HttpServletResponse resp)throws
ServletException,IOException
{
resp.setContentType("text/html");
PrintWriter out=resp.getWriter();
String tilte="using GET method to read from data";
out.println("<br>Name</br>:"+req.getParameter("name"));
out.println("<br>RollNo</br>:"+req.getParameter("rollnumber"));
out.println("<br>Age</br>:"+req.getParameter("age"));
out.println("<br>Sex</br>:"+req.getParameter("sex"));
out.println("<br>Mark</br>:"+req.getParameter("mark"));
out.println("<br>Qualification</br>:"+req.getParameter("qualification"));
}
}
```

Hello.html

```
<html>
<body>
<form action=http://localhost:8080/webapps/sample/WEB-INF/classes/helloform
method="GET">
name:<input type="text" name="Name"><br><br>
RollNo:<input type="text" rollnumber="rollnumber><br><br>
sex:<input type="text" sex="sex"><br><br>
Age:<input type="text" age="age"><br><br>
```

```
Mark:<input type="text" mark="mark"><br><br>
qualification::<input type="text" qualification="qualification"><br><br>
<input type="submit" values="submit">
</form>
</body>
</html>
```

OUTPUT: