1. EMPLOYEE

AIM

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

PROGRAM

```
import java.io.*;
import java.sql.*;
class employee
{
 Connection con;
 BufferedReader br;
 PreparedStatement pst;
 ResultSet rs;
 void load()
 {
  try
  {
   Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
   con=DriverManager.getConnection("jdbc:odbc:empdb","","");
  }
  catch(ClassNotFoundException e)
  {
   System.out.println("Driver Error"+e.getMessage());
  }
  catch(SQLException d)
  {
   System.out.println("SQL Error"+d.getMessage());
  }
```

```
}
void insertion()throws IOException
{
 try
 {
  br=new BufferedReader(new InputStreamReader(System.in));
  pst=con.prepareStatement("insert into employee values(?,?,?,?)");
  System.out.println("enter empid:");
  pst.setInt(1,Integer.parseInt(br.readLine()));
  System.out.println("enter emp name: ");
  pst.setString(2,br.readLine());
  System.out.println("enter emp designation ");
  pst.setString(3,br.readLine());
  System.out.println("enter emp salary:");
  pst.setInt(4,Integer.parseInt(br.readLine()));
  pst.executeUpdate();
  con.close();
  pst.close();
 }
 catch(SQLException e2)
 {
  System.out.println("SQL Error"+e2.getMessage());
 }
}
void display()throws IOException
{
 try
 {
```

```
 br=new BufferedReader(new InputStreamReader(System.in));
 pst=con.prepareStatement("select * from employee where empid=?");
 System.out.println("enter empid:");
 pst.setInt(1,Integer.parseInt(br.readLine()));
 rs=pst.executeQuery();
 System.out.println("details of employee ");
 System.out.println("\nEID\tENAME\t\tEDESI\t\tESAL\n");
 System.out.println("\n\n");
 while(rs.next())
 {
  System.out.print(rs.getInt("empid")+"\t");
  System.out.print(rs.getString("empname")+"\t\t");
  System.out.print(rs.getString("empdesi")+"\t");
  System.out.print(rs.getInt("empsal")+"\t");
 }
 pst.close();
 con.close();
}
catch(SQLException e3)
{
 System.out.println("SQL Error"+e3.getMessage());
}
}
void updation()throws IOException
{
 try
 {
  br=new BufferedReader(new InputStreamReader(System.in));
```

```
  pst=con.prepareStatement("update employee set empid=?,empname=?,empdesi=?,empsal=?
where empid=?");
  System.out.println("enter empid:");
  pst.setInt(5,Integer.parseInt(br.readLine()));
  System.out.println("enter new empid:");
  pst.setInt(1,Integer.parseInt(br.readLine()));
  System.out.println("enter new empname");
  pst.setString(2,br.readLine());
  System.out.println("enter new emp designation ");
  pst.setString(3,br.readLine());
  System.out.println("enter new emp salary");
  pst.setInt(4,Integer.parseInt(br.readLine()));
  pst.executeUpdate();
  System.out.println("UPDATED SUCCESFULLY");
  con.close();
  pst.close();
 }
 catch(SQLException e4)
 {
  System.out.println("SQL Error"+e4.getMessage());
 }
}
 void deletion() throws IOException
 {
  try
  {
   br=new BufferedReader(new InputStreamReader(System.in));
   pst=con.prepareStatement("delete * from employee where empid=?");
   System.out.println("enter emp id");
```

```
 pst.setInt(1,Integer.parseInt(br.readLine()));
 pst.executeUpdate();
 System.out.println("DELETE SUCCESFULLY");
 }
 catch(SQLException e5)
 {
  System.out.println("SQL Error"+e5.getMessage());
 }
}
public static void main(String args[]) throws IOException
{
 employee obj=new employee();
 BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
 int choice;
 do
 {
  System.out.println("\n\nEMPLOYEE INFORMATION SYSTEM\n");
  System.out.println("\n 1:Insertion\n 2:Display\n 3:Updation\n 4:Deletion\n 5:Exit\n");
  System.out.println("ENTER YOUR CHOICE");
  choice=Integer.parseInt(br.readLine());
  switch(choice)
  {
   case 1: obj.load();
           obj.insertion();
           break;
   case 2: obj.load();
           obj.display();
           break;
```

```
    case 3: obj.load();
            obj.updation();
            break;
    case 4: obj.load();
            obj.deletion();
            break;
    case 5: System.exit(0);
            break;
    default:
            System.out.println("invalid choice");
   }
  }while(choice!=5);
 }
}
```

OUTPUT

EMPLOYEE INFORMATION SYSTEM

1:Insertion

2:Display

3:Updation

4:Deletion

5:Exit

ENTER YOUR CHOICE

1

enter empid:

3

enter emp name:

hari

enter emp designation

clerk

enter emp salary:

2000

EMPLOYEE INFORMATION SYSTEM

1:Insertion

2:Display

3:Updation

4:Deletion

5:Exit

ENTER YOUR CHOICE

2

enter empid:

3

details of employee

EID ENAME EDESI ESAL

3 hari clerk 2000

EMPLOYEE INFORMATION SYSTEM

1:Insertion

2:Display

3:Updation

4:Deletion

5:Exit

ENTER YOUR CHOICE

3

enter empid:

3

enter new empid:

4

enter new empname

ram

enter new emp designation

manager

enter new emp salary

10000

UPDATED SUCCESFULLY

EMPLOYEE INFORMATION SYSTEM

1:Insertion

2:Display

3:Updation

4:Deletion

5:Exit

ENTER YOUR CHOICE

2

enter empid:

4

details of employee

EID ENAME EDESI ESAL

4 ram manager 10000

EMPLOYEE INFORMATION SYSTEM

1:Insertion

2:Display

3:Updation

4:Deletion

5:Exit

ENTER YOUR CHOICE

4

enter emp id

4

DELETE SUCCESFULLY

EMPLOYEE INFORMATION SYSTEM

1:Insertion

2:Display

3:Updation

4:Deletion

5:Exit

ENTER YOUR CHOICE

5

2. STUDENT DETAILS

AIM

JDBC program to connect to Student table. Implement the record scrolling functions – first(), last(), next(), previous(), beforeFirst(), afterLast(), absolute() .

PROGRAM

```
import java.sql.*;
import java.io.*;
public class Student
{
public static void main(String args[])throws IOException
{
int ch,row;
BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
try
{
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
}
catch(ClassNotFoundException e)
{
System.out.println("unable to load driver");
}
try
{
Connection con=DriverManager.getConnection("jdbc:odbc:sdb","","");
Statement
stmt=con.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE,ResultSet.CONCUR_UPDA
TABLE);
ResultSet rs=stmt.executeQuery("select * from student");
```

```
System.out.println("current table");
System.out.println("SNO\t SNAME\tMARKS\n");
while(rs.next())
{
System.out.println(rs.getInt("sno")+"\t"+rs.getString("sname")+"\t"+rs.getInt("marks"));
}
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 last\n6.exit\nenter your choice:");
ch=Integer.parseInt(br.readLine());
switch(ch)
{
case 1:rs.first();
System.out.println("\nmoving to the first row");
System.out.println(rs.getInt("sno")+"\t"+rs.getString("sname")+"\t"+rs.getInt("marks"));
break;
case 2:rs.next();
System.out.println("\nmoving to the next row");
System.out.println(rs.getInt("sno")+"\t"+rs.getString("sname")+"\t"+rs.getInt("marks"));
break;
case 3:rs.previous();
System.out.println("\nmoving to the previous row");
System.out.println(rs.getInt("sno")+"\t"+rs.getString("sname")+"\t"+rs.getInt("marks"));
break;
case 4:System.out.println("enter the row number");
row=Integer.parseInt(br.readLine());
rs.absolute(row);
System.out.println("\nmoving to the specified row");
```

```
System.out.println(rs.getInt("sno")+"\t"+rs.getString("sname")+"\t"+rs.getInt("marks"));
break;
case 5: rs.last();
System.out.println("\nmoving to the last row");
System.out.println(rs.getInt("sno")+"\t"+rs.getString("sname")+"\t"+rs.getInt("marks"));
break;
case 6:
System.exit(0);
}
}
while(ch!=6);
rs.close();
stmt.close();
con.close();
}
catch(SQLException e)
{
System.out.println(" connection failed"+e.getMessage());
e.printStackTrace(System.out);
}
}
}
```

OUTPUT

current table

SNO SNAME MARKS

101 nikhila 89

102 Arya 98

103 Anu 88

104 Aiswarya 95

MENU

1.move to first

2.move to next

3.move to previous

4.move to specified row

5.move to last

6.exit

enter your choice:

1

moving to the first row

101 nikhila 89

MENU

1.move to first

2.move to next

3.move to previous

4.move to specified row

5.move to last

6.exit

enter your choice:

2

moving to the next row

102 Arya 98

MENU

1.move to first

2.move to next

3.move to previous

4.move to specified row

5.move to last

6.exit

enter your choice:

3

moving to the previous row

101 nikhila 89

MENU

1.move to first

2.move to next

3.move to previous

4.move to specified row

5.move to last

6.exit

enter your choice:

4

enter the row number

2

moving to the specified row

102 Arya 98

MENU

1.move to first

2.move to next

3.move to previous

4.move to specified row

5.move to last

6.exit

enter your choice:

5

moving to the last row

104 Aiswarya 95

MENU

1.move to first

2.move to next

3.move to previous

4.move to specified row

5.move to last

6.exit

enter your choice: 6

3. DATABASE METADATA

AIM

JDBC program to display database metadata**.**

PROGRAM

```
import java.sql.*;
class Dbmd
{
public static void main(String args[])
{
try
{
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
Connection con = DriverManager.getConnection ("jdbc:odbc:Employeedb"," "," ");
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("Database Product Name: "+dbmd.getDatabaseProductName());
System.out.println("Database Product Version: "+dbmd.getDatabaseProductVersion());
con.close();
}
catch(Exception e)
{
System.out.println(e);
```

```
}

}

}
```

OUTPUT

DriverName: JDBC-ODBC Bridge (ACEODBC.DLL)

DriverVersion: 2.0001 (Microsoft Office 2007 Access database engine)

UserName: admin

Database Product Name: ACCESS

Database Product Version: 12.00.0000

4. RESULTSET METADATA

AIM

JDBC program to display Resultset metadata.

PROGRAM

```
import java.sql.*;
import java.util.StringTokenizer;
public class Rsmd
{
public static void main(java.lang.String[] args)
 {
System.out.println("--- Table Viewer ---");
try
 {
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
Connection con = DriverManager.getConnection   ("jdbc:odbc:Employeedb"," "," ");
Statement stmt = con.createStatement();
ResultSet rs = stmt.executeQuery("select * from employee");
ResultSetMetaData rsmd = rs.getMetaData();
int columnCount = rsmd.getColumnCount();
for(int col = 1; col <= columnCount; col++)
{
System.out.print(rsmd.getColumnLabel(col));
System.out.print("("+rsmd.getColumnTypeName(col)+")");
if(col < columnCount)
System.out.print(", ");
}
System.out.println();
```

```
while(rs.next())
{
for(int col = 1; col <= columnCount; col++)
{
System.out.print(rs.getString(col));
if(col < columnCount)
System.out.print(", ");
}
System.out.println();
}
rs.close();
stmt.close();
con.close();
}
catch (ClassNotFoundException e)
{
System.out.println("Unable to load database driver class");
}
catch (SQLException e)
{
System.out.println("SQL Exception:"+ e.getMessage());
}
}
}
```

OUTPUT

-- Table Viewer ---

ID(INTEGER), name(VARCHAR), place(VARCHAR)

1, Binesh, Chepparamba

2, Aiswarya, Kottur

5. COMPLEX NUMBER OPERTATION

AIM

RMI program to perform complex number operations.

PROGRAM

File:complex.java

```
import java.io.*;
public class complex implements Serializable
{
 int real,image;
public complex()
{
  real=image=0;
}
public complex(int i,int j)
{
 real=i;image=j;
}
public void display()
{
 System.out.println(real+" + "+image+" i ");
}
public void read()throws IOException
{
  BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
System.out.print("\nEnter real part: ");
real=Integer.parseInt(br.readLine());
 System.out.print("\nEnter img part: ");
```

```
image=Integer.parseInt(br.readLine());
}
}
```

File:compinterface.java

```
import java.rmi.*;
public interface compinterface extends Remote
{
public complex add(complex a,complex b)throws RemoteException;
}
```

File:compimpl.java

```
import java.rmi.*;
import java.rmi.server.*;
public class compimpl extends UnicastRemoteObject implements compinterface
{
public compimpl()throws RemoteException
{
super();
}
public complex add(complex a,complex b)throws RemoteException
{
a.real+=b.real;
a.image+=b.image;
return a;
}
}
```

File:compreg.java

```
import java.rmi.*;
public class compreg
{
public static void main(String[] args)
{
try
{
 compimpl obj=new compimpl();
 Naming.rebind("ComplexNumber",obj);
System.out.println("object registered");
}
catch(Exception e)
{
e.printStackTrace();
}
}
}
```

File:compclient.java

```
import java.rmi.*;
import java.io.*;
public class compclient
{
public static void main(String[] args)throws IOException
{
try
```

```
{
  complex ob1=new complex();
  complex ob2=new complex();
 System.out.print("\nEnter first complex number\n");
 ob1.read();
 System.out.print("\nEnter second complex number\n");
 ob2.read();
 compinterface cmp=(compinterface)Naming.lookup("rmi://127.0.0.1/ComplexNumber");
 complex ob3=cmp.add(ob1,ob2);
 System.out.print("\nFirst  complex number    :");
ob1.display();
System.out.print("\nSecond  complex number    :");
ob2.display();
System.out.print("\nResultant  complex number  :");
ob3.display();
}
catch(RemoteException e)
{
}
catch(NotBoundException e)
{
}
}
}
```

OUTPUT

Enter first complex number

Enter real part: 3

Enter img part: 5

Enter second complex number

Enter real part: 8

Enter img part: 4

First complex number :3 + 5 i

Second complex number :8 + 4 i

Resultant complex number :11

6. BANK OPERATION

AIM

RMI Program to perform bank operations.

PORGRAM

File:Account.java

```
import  java.rmi.Remote;
import  java.rmi.RemoteException;
public interface Account extends java.rmi.Remote
{
public String getName()throws RemoteException;
public float getBal() throws RemoteException;
public float withdrow(float amt)throws RemoteException;
public float deposit(float amt)throws RemoteException;
}
```

File:AccountImpl.java

```
import java.rmi.server.UnicastRemoteObject;
import java.rmi.RemoteException;
public class AccountImpl extends UnicastRemoteObject implements Account
{
private float mbal;
private String mname="";
public AccountImpl(String name)throws RemoteException
{
mname=name;
mbal=1000;
}
public String  getName()throws RemoteException
```

```
{
return mname;
}
public float getBal()throws RemoteException
{
return mbal;
}
public float withdrow(float amt)throws RemoteException
{
if(mbal-amt>=1000)
{
mbal=mbal-amt;
return 0;
}
else
return -1;
}
public float deposit(float amt)throws RemoteException
{
mbal=mbal+amt;
return mbal;
}
}
```

File:AccountReg.java

```
import java.rmi.Naming;
public class AccountReg
{
public static void main(String args[])
{
try
{
AccountImpl ob1=new AccountImpl("jack");
Naming.rebind("jack",ob1);
System.out.println("registered Account");
}
catch(Exception e)
{
e.printStackTrace();
}
}
}
```

File:AccountClient.java

```
import java.io.*;
import java.rmi.Naming;
public class AccountClient
{
public static void main(String args[])throws IOException
{
String name;
int ch;
float amount,bal,bal1;
```

```
try
{
Account obj=(Account)Naming.lookup("rmi://127.0.0.1/jack");
BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
do
{
System.out.println("MENU\n1.Deposit\n2.Withdrow\n3.display\n4.exit\n enter ur choice");
ch=Integer.parseInt(br.readLine());
switch(ch)
{
case 1:
System.out.print("enter the amount to be deposited");
amount=Float.parseFloat(br.readLine());
bal=obj.deposit(amount);
System.out.println("deposited successfully");
System.out.println("new balance:"+bal);
break;
case 2:
System.out.print("enter the amount to be withdtrown");
amount=Float.parseFloat(br.readLine());
bal=obj.withdrow(amount);
if(bal!=-1)
{
bal1=obj.getBal();
System.out.println("withdrown successfulll\nNew balance:"+bal1);
}
else
System.out.println("no sufficient balance");
```

```
break;
case 3:
System.out.println("name:"+obj.getName());
System.out.println("current balance"+obj.getBal());
break;
case 4:
System.exit(0);
default:System.out.println("enter a valid choice");
}
}
while(ch!=4);
}
catch(Exception e)
{
System.out.print("Error:"+e);
}
}
}
```

OUTPUT

MENU

1.Deposit

2.Withdrow

3.display

4.exit

enter ur choice

1

enter the amount to be deposited 1000

deposited successfully

new balance:2000.0

MENU

1.Deposit

2.Withdrow

3.display

4.exit

enter ur choice

2

enter the amount to be withdtrown 1000

withdrown successfulll

New balance:1000.0

MENU

1.Deposit

2.Withdrow

3.display

4.exit

enter ur choice

3

name:jack

current balance1000.0

MENU

1.Deposit

2.Withdrow

3.display

4.exit

enter ur choice

4

7. MATRIX OPERATION

AIM

RMI Program to perform matrix operations.

PORGRAM

File:MatrixInter.java

```
import java.rmi.Remote;
import java.rmi.RemoteException;
public interface MatrixInter extends java.rmi.Remote
{
public Matrix add(Matrix a,Matrix b)throws RemoteException;
public Matrix sub(Matrix a,Matrix b)throws RemoteException;
}
```

File:MatrixImpl.java

```
import java.rmi.server.UnicastRemoteObject;
import java.rmi.RemoteException;
public class MatrixImpl extends UnicastRemoteObject implements MatrixInter
{
public Matrix m;
public MatrixImpl(int a,int b)throws RemoteException
{
m=new Matrix(a,b);
}
public MatrixImpl()throws RemoteException{}
public Matrix add(Matrix m1,Matrix m2)throws RemoteException
{
Matrix m3=new Matrix(m1.r,m1.c);
```

```
for(int i=0;i<m1.r;i++)
for(int j=0;j<m1.c;j++)
m3.a[i][j]=m1.a[i][j]+m2.a[i][j];
return m3;
}
public Matrix sub(Matrix m1,Matrix m2)throws RemoteException
{
Matrix m3=new Matrix(m1.r,m1.c);
for(int i=0;i<m1.r;i++)
for(int j=0;j<m1.c;j++)
m3.a[i][j]=m1.a[i][j]-m2.a[i][j];
return m3;
}
}
```

File:MatrixReg.java

```
import java.rmi.Naming;
public class MatrixReg
{
public static void main(String args[])
{
try
{
MatrixImpl mat=new MatrixImpl();
Naming.rebind("matrix",mat);
System.out.println("Object Registered");
}
catch(Exception e)
```

```
{
e.printStackTrace();
}
}
}
```

File:MatrixClient.java

```
import java.rmi.Naming;
import java.io.*;
public class MatrixClient
{
public static void main(String args[])throws IOException
{
BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
int r1,c1,r2,c2;
int ch;
System.out.println("Enter number of rows and columns of 1st matrix");
r1=Integer.parseInt(br.readLine());
c1=Integer.parseInt(br.readLine());
Matrix a1=new Matrix(r1,c1);
System.out.println("Enter number of rows and columns of 2nd matrix");
r2=Integer.parseInt(br.readLine());
c2=Integer.parseInt(br.readLine());
Matrix a2=new Matrix(r2,c2);
System.out.println("enter elements of 1st matrix");
a1.read();
System.out.println("enter elements of 2nd matrix");
a2.read();
```

```
Matrix a3;
if((r1==r2)&&(c1==c2))
{
try
{
MatrixInter obj=(MatrixInter)Naming.lookup("rmi://localhost/matrix");
System.out.println("\nMatrices are");
a1.display();
System.out.println();
a2.display();
do
{
System.out.println("\nMENU\n1.add\n2.subtract\n3.exit\nenter choice");
ch=Integer.parseInt(br.readLine());
switch(ch)
{
case 1:a3=obj.add(a1,a2);
System.out.println("\nsum of matrices");
a3.display();
break;
case 2:a3=obj.sub(a1,a2);
System.out.println("\ndifference of matrices");
a3.display();
break;
case 3:System.exit(0);
default:
System.out.println("\nenter a valid choice");
}
```

```
}while(ch!=3);
}
catch(Exception e)
{
System.out.println("Error"+e);
}
}
else
System.out.println("\nOperations cannot be performed");
}
}
```

File:Matrix.java

```
import java.io.*;
import java.io.Serializable;
public class Matrix implements Serializable
{
public int[][]a=new int[5][5];
public int r,c;
public Matrix(int x,int y)
{
r=x;c=y;
}
public void read()throws IOException
{
BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
for(int i=0;i<r;i++)
for(int j=0;j<c;j++)
```

```
a[i][j]=Integer.parseInt(br.readLine());
}
public void display()
{
for(int i=0;i<r;i++)
{
System.out.println();
for(int j=0;j<c;j++)
System.out.println(a[i][j]+"\t");
}
}
}
```

OUTPUT

Enter number of rows and columns of 1st matrix

2

2

Enter number of rows and columns of 2nd matrix

2

2

enter elements of 1st matrix

2

2

2

2

enter elements of 2nd matrix

1

1

1

1

Matrices are

2

2

2

2

1

1

1

1

MENU

1.add

2.subtract

3.exit

enter choice

1

sum of matrices

3

3

3

3

MENU

1.add

2.subtract

3.exit

enter choice

2

difference of matrices

1

1

1

1

MENU

1.add

2.subtract

3.exit

enter choice

3

8. STUDENT DETAILS

AIM

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

PROGRAM

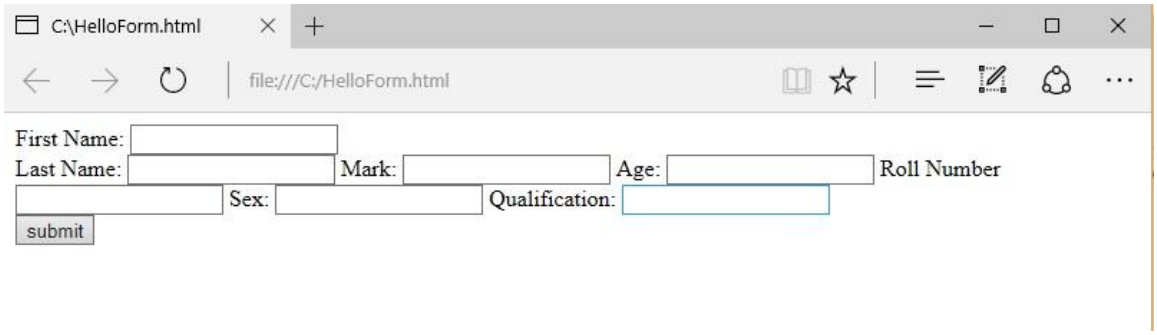
//HelloForm.html

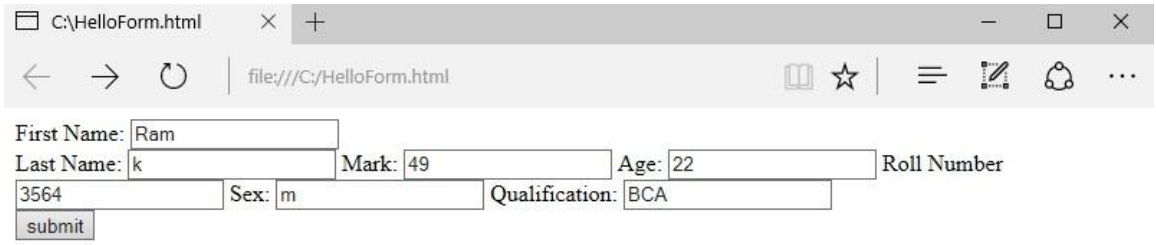
```
<html>
<body>
<form action=http://localhost:8080/Lab/HelloForm method="GET">
First Name: <input type="text" name="first_name"></br>
Last Name: <input type="text" name="last_name">
Mark: <input type="text" mark="mark">
Age: <input type="text" age="age">
Roll Number <input type="text" roll_number="roll_number">
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





FirstName
:Ram
LastName
:k
Mark
:49
Age
:22
Roll_Number:3564
Sex
:m
Qualification
:BCA

9. REQUEST INORMATION

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:

```
// ReqInf.java

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


localhost
localhost:8080/Lab/ReqInf
Url:http://localhost:8080/Lab/ReqInf
Scheme:http
ServerName:localhost
HostName:0:0:0:0:0:0:0:1
Port:8080
Method:GET
Host:0:0:0:0:0:0:0:1

10. ATM SERVLET

AIM

Servlet program to show basic bank operations

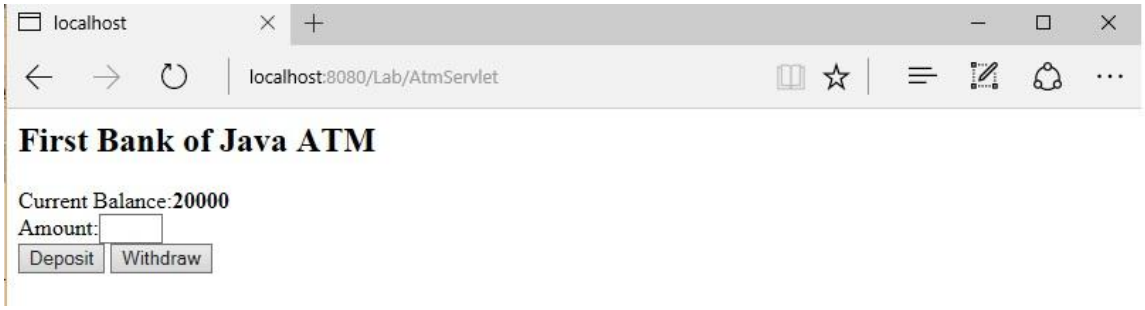
PROGRAM

```
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


localhost
localhost:8080/Lab/AtmServlet
First Bank of Java ATM
Current Balance:20000
Amount:
Deposit
Withdraw

11. FILE SERVLET

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

ReadTextFile.html

```
<BODY>
<H2>Getting File contents from the Server</H2>
<FORM METHOD="get" ACTION="http://localhost:8080/Lab/TextFileRead">
Enter File Name<INPUT TYPE="text" NAME="filename"><BR>
<INPUT TYPE="submit"  VALUE="Send me">
</FORM>
</BODY>
```

TextFileRead.java

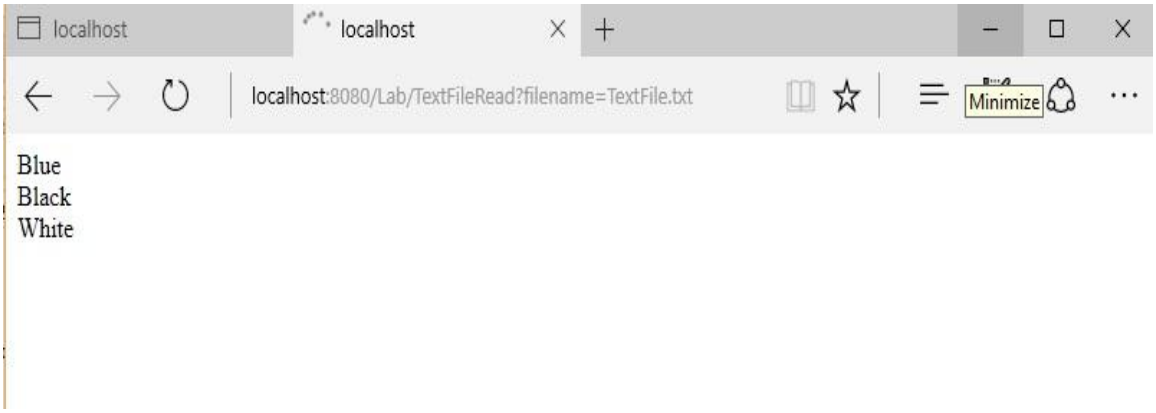
```
import javax.servlet.ServletException;
import javax.servlet.http.*;
import java.io.*;
public class TextFileRead extends HttpServlet
{
        public void service(HttpServletRequest req,HttpServletResponse res)throws
ServletException,IOException
        {
                res.setContentType("text/html");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

```
                {
                        pw.println(str+"<BR>");
                }
                br.close();
                pw.close();
        }
}
```

OUTPUT


TextFile - Notepad
File Edit Format View Help
Blue
Black
White


localhost
C:\TextFileRead.html
file:///C:/TextFileRead.html
Getting File contents from the Server
Enter File Name TextFile.txt
Send me


localhost
localhost
localhost:8080/Lab/TextFileRead?filename=TextFile.txt
Minimize
Blue
Black
White

12. SESSION HANDLING

AIM

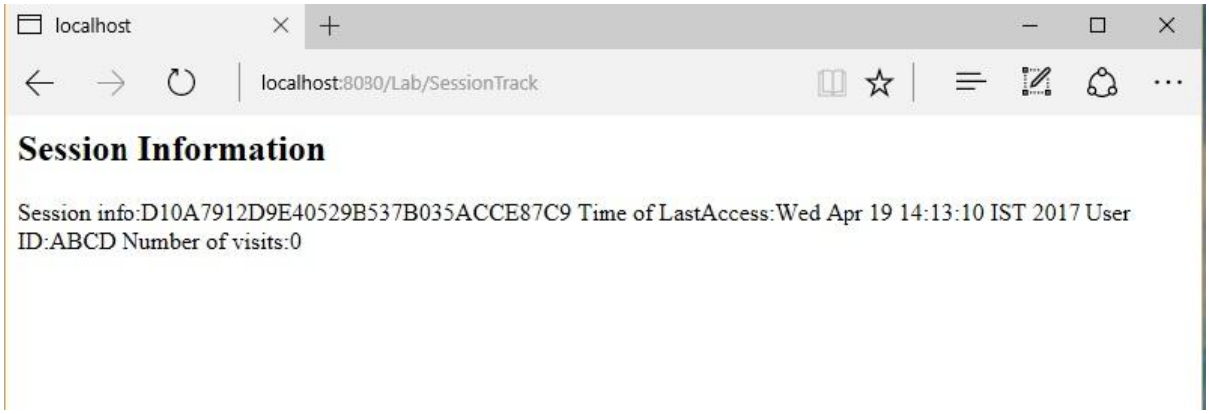
Session handling servlet that displays total number of visits to that page.

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


localhost
localhost:8080/Lab/SessionTrack
Session Information
Session info:D10A7912D9E40529B537B035ACCE87C9 Time of LastAccess:Wed Apr 19 14:13:10 IST 2017 User ID:ABCD Number of visits:0