**WEEK-1**

**AIM:**

**Write a java program that works as a simple calculator. Use a grid layout to arrange buttons for the digits and for the +, -, \*, % operations. Add a text field to display the result. Handle any possible exceptions like divided by zero.**

**PROGRAM:**

Import java.awt.\*;

Import java.awt.event.\*;

import java.applet.\*;

public class Calculator extends Applet implements ActionListener

{

String msg=" ";

**int** p,q,result;

TextField t1;

Button b[]=**new** Button[10];

Button add,sub,mul,div,clear,mod,EQ;

charop;

**public void** init()

{

t1=**new** TextField(10);

GridLayout gl=**new** GridLayout(4,5);

setLayout(gl);

**for**(**int** i=0;i<10;i++)

{

b[i]=**new** Button(""+i);

}

add=**new** Button("add");

sub=**new** Button("sub");

mul=**new** Button("mul");

div=**new** Button("div");

mod=**new** Button("mod");

clear=**new** Button("clear");

EQ=**new** Button("EQ");

t1.addActionListener(**this**);

add(t1);

**for**(**int** i=0;i<10;i++)

{

add(b[i]);

}

add(add);

add(sub);

add(mul);

add(div);

add(mod);

add(clear);

add(EQ);

**for**(**int** i=0;i<10;i++)

{

b[i].addActionListener(**this**);

}

add.addActionListener(**this**);

sub.addActionListener(**this**);

mul.addActionListener(**this**);

div.addActionListener(**this**);

mod.addActionListener(**this**);

clear.addActionListener(**this**);

EQ.addActionListener(**this**);

}

**public void** actionPerformed(ActionEvent ae)

{

String str=ae.getActionCommand();

**char** ch=str.charAt(0);

**if** ( Character.*isDigit*(ch)) t1.setText(t1.getText()+ str);

**else if**(str.equals("add"))

{

p=Integer.*parseInt*(t1.getText());

OP='+';

t1.setText("");

}

**else if**(str.equals("sub"))

{

p=Integer.*parseInt*(t1.getText());

OP='-';

t1.setText("");

}

**else if**(str.equals("mul"))

{

p=Integer.*parseInt*(t1.getText());

OP='\*';

t1.setText("");

}

**else if**(str.equals("div"))

{

p=Integer.*parseInt*(t1.getText());

OP='/';

t1.setText("");

}

**else if**(str.equals("mod"))

{

p=Integer.*parseInt*(t1.getText());

OP='%';

t1.setText("");

}

**if**(str.equals("EQ"))

{

q=Integer.*parseInt*(t1.getText());

**if**(OP=='+')

result=p+q;

**else if**(OP=='-')

result=p-q;

**else if**(OP=='\*')

result=p\* q;

**else if**(OP=='/')

result=p/q;

**else if**(OP=='%')

result=p%q;

t1.setText(""+result);

}

**if**(str.equals("clear"))

{

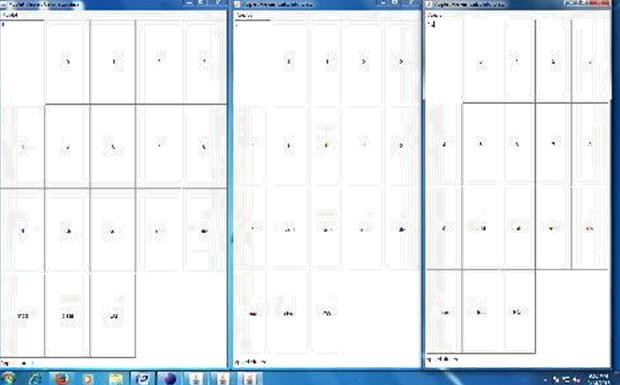
t1.setText("");

}

}

}

# Output:

****

**WEEK 2**

**AIM:**

**Write an applet that displays a simple message.**

**PROGRAM:**

import java.applet.Applet;

import java.awt.Graphics;

public class Hello extends Applet

{

public void paint(Graphics g)

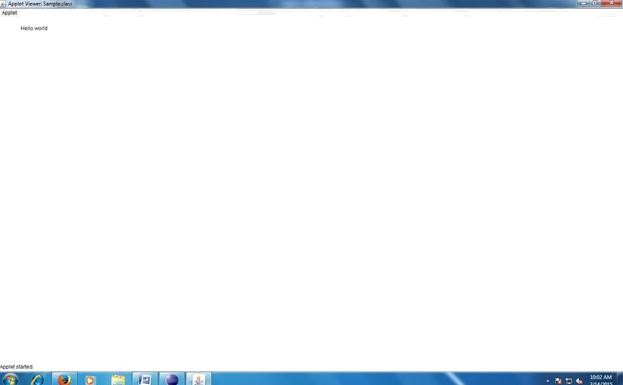
{

g.drawString("Hello world",50,30);

}

}

**OUTPUT:**



**Develop an applet that receives an integer in one text field, and computes its factorial**

**Value and returns it in another text field, when the button named “Compute” is clicked.**

**PROGRAM:**

import java.awt.\*;

import java.awt.event.\*;

public class factorial extends java.applet.Applet implements ActionListener

{

TextField t1,t2;

Label l1,l2,l3;

Button b1;

int fact=1,n,i;

factorial e;

public void init()

{

e=this;

t1=new TextField(10);

t2=new TextField(10);

l1=new Label("factorial of a number");

l2=new Label("enter number");

l3=new Label("result");

b1=new Button("compute");

add(l1);

add(l2);

add(l3);

add(t1);

add(t2);

add(b1);

b1.addActionListener(this);

}

public void actionPerformed(ActionEvent ae)

{

String str=t1.getText();

n=Integer.parseInt(str);

for(i=n;i>1;i--)

{

fact=fact\*i;

}

String msg=""+fact;

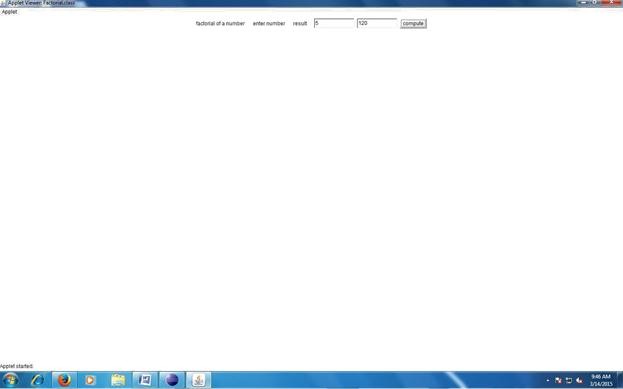
t2.setText(msg);

fact=1;

}

}

**Output:**



**WEEK3**

**AIM:**

**Write a program that creates a user interface to perform integer divisions. The user enters two numbers in the textfields, Num1 and Num2. The division of Num1 and Num2 is displayed in the Result field when the Divide button is clicked. If Num1 or Num2 were not an integer, the program would throw a NumberFormatException. If Num2 were Zero, the program would throw an ArithmeticException Display the exception in a message dialog box.**

**PROGRAM:**

import java.awt.\*;

import javax.swing.\*;

import java.applet.Applet;

import java.awt.event.\*;

public class Division extends Applet implements ActionListener

{

TextField t1,t2,t3;

Button b;

Label L1,L2,L3,L4;

String s;

Division e;

public void init()

{

e=this;

t1=new TextField(10);

t2=new TextField(10);

t3=new TextField(10);

L1=new Label("enter num1");

L2=new Label("enter num2");

L3=new Label("Result is");

L4=new Label("Division of 2numbers");

b=new Button("Divide");

add(L4);

add(L1);

add(t1);

add(L2);

add(t2);

add(L3);

add(t3);

add(b);

b.addActionListener(this);

}

public void actionPerformed(ActionEvent ae)

{

try

{

int num1=Integer.parseInt(t1.getTex t());

intnum2=Integer.parseInt(t2.getTex t());

s=""+(num1/num2);

t3.setText(s);

}

catch(ArithmeticException a)

{

JOptionPane.showMessageDialog(null,"Divide by zero");

}

catch(NumberFormatException b)

{

JOptionPane.showMessageDialog(null,"NumberFormateException");

}

}

**Output:**



**WEEK4**

**AIM:**

**Write a java program that implements a multi-thread applications that has three threads. First thread generates random integer every 1 second and if the value is even, second thread computes the square of the number and prints. If the value is odd, the thread will print the value of the number.**

**PROGRAM:**

package Mthread;

import java.util.Random;

public class Mthread

{

public static void main(String args[])

{

A a=new A("one");

a.start();

}

}

class A extends Thread

{

String tname; Random r; Thread t1,t2; A(String x)

{

this.tname=x;

}

public void run()

{

try

{

int num=0;

r=new Random();

num=r.nextInt(1 00);   
for(int i=0;i<10;i++)

{

if(num%2==0)

{

t1=new Thread(new even(num));

t1.start();

}

else

{

t2=new Thread(new odd(num));

t2.start();

}

Thread.sleep(1000);

}

}

catch(InterruptedException e)

{

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

}

Catch(Exception a)

{

System.out.println("Exception is"+a);

} } }

class even implements Runnable

{

int x;

even(int x)

{

This.x=x;

}

public void run()

{

System.out.println(“num is even”+x+”its square is”+(x\*x));

}}

class odd implements Runnable

{

int x; odd(int x)

{

this.x=x;

}

public void run()

{

System.out.println(“num is odd”+x+”its cube is”+(x\*x\*x));

}

}

**OUTPUT:**

num is odd 69 it is cubed 328509

num is odd 13 it is cubed 2197

num is odd 23 it is cubed 12167

num is even 32 it is squared 1024

num is odd 91 it is cubed 753571

num is odd 47 it is cubed 103823

num is odd 33 it is cubed 35937

num is even 24 it is squared 576

num is even 80 it is squared 6400

num is even 42 it is squared 1764

num is even 68 it is squared 4624

num is even 2 it is squared 4892

num is even 84 it is squared 7056

num is even 70 it is squared 4900

num is even 60 it is squared 3600

num is even 84 it is squared 7056 num

is odd 51 it is cubed 132651

num is even 60 it is squared 3600

# WEEK5

**AIM:**

**Write a java program that connects to a database using JDBC and does add, delete, modify and retrieve operations.**

**PROGRAM:**

import java.sql.\*;

public class JdbcExample

{

static final String JDBC\_DRIVER="com.mysql.jdbc.Driver";

static final String DB\_URL="jdbc:mysql://localhost/jdbc";

static final String USER="root";

static final String PASS="Gcet@05";

public static void main(String args[])

{

Connection conn=null;

Statement stmt=null;

try

{

System.out.println("connecting to database--");

conn=DriverManager.getConnection(DB\_URL,USE R,PASS);

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

stmt=conn.createStatement();

String sql,sql1,sql2,sql3;

Sql1=”insert into employee values(23,’bob’,’s’,20)”;

int s1= stmt.executeUpdate(sql1);

sql2=”update employee set age=18 where id=3”;

int s2= stmt.executeUpdate(sql2);

sql3=”delete from employee where id=24”;

int s3= stmt.executeUpdate(sql3);

sql="SELECT id,first,last,age FROM employee";

ResultSet rs=stmt.executeQuery(sql);

while(rs.next())

{

int id=rs.getInt("id");

int age=rs.getInt("age");

String first=rs.getString("first");

String last=rs.getString("last");

System.out.println("ID:"+id);

System.out.println("Age:"+age);

System.out.println("First:"+first);

System.out.println("Last:"+last);

}

rs.close();

stmt.close();

conn.close();

}

catch(SQLException se)

{

se.printStackTrace();

}

System.out.println("Goodbye");

}

}

**OUTPUT:**

connecting to database— creating

|  |  |  |  |
| --- | --- | --- | --- |
| statement— | | | |
| id:2 | first:siri | last:  m | age:2  3 |
| id:3 | first:deep | last:v | age:1 8 |
| id:23 | first:bob | last:s | age:2  0 |

Goodbye

# WEEK6

**AIM:**

**Write a java program that simulates a traffic light. The program lets user select one of the three lights: red, yellow, or, green with radio buttons.**

**On selecting a button, an appropriate message with “stop” initially, there is no message shown.**

**PROGRAM:**

import java.applet.Applet;

import java.awt.Checkbox; i

mport java.awt.CheckboxGroup;

import java.awt.Color;

import java.awt.Graphics;

import java.awt.event.ItemEvent;

import java.awt.event.ItemListener;

public class Trafficlights extends Applet implements ItemListener

{

String msg="";

Checkbox red,yellow,green;

CheckboxGroup cg=null;

public void init()

{

cg=new CheckboxGroup();

Checkbox red=new Checkbox("red",cg,true);

red.setBackground(Color.red);

Checkbox yellow=new Checkbox("yellow",cg,false);

yellow.setBackground(Color.yellow);

Checkbox green=new Checkbox("green",cg,false);

green.setBackground(Color.green);

add(red);

add(yellow);

add(green);

red.addItemListener(this);

yellow.addItemListener(this);

green.addItemListener(this);

}

public void itemStateChanged(ItemEvent ie)

{

repaint();

}

public void paint(Graphics g)

{

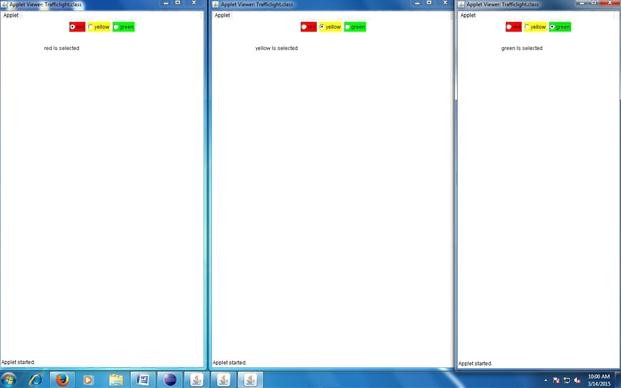
Checkbox chk=cg.getSelectedCheckbox();

g.drawString(chk.getLabel()+" Is selected",101,70);

}

}

# OUTPUT:



# WEEK7

**AIM:**

**Write a java program to create an abstract class named shape that contains two integers and an empty method named printArea(). Provide three classes named Rectangle, Triangle and Circle such that classes contains only the method printArea() that prints the area of the given shape.**

**PROGRAM**:

import java.util.\*;

abstract class shape

{

int x,y;

abstract void area(double x,double y);

}

class Rectangle extends shape

{

void area(double x,double y)

{

System.*out*.println("area of rectangle:"+(x\*y));

}

}

class Circle extends shape

{

void area(double x,double y)

{

System.*out*.println("area of circle:"+(3.14\*x\*x));

}

}

class Triangle extends shape

{

void area(double x,double y)

{

System.*out*.println("area of triangle:"+(0.5\*x\*y));

}

}

public class AbstactDDemo

{

public static void main(String[] args)

{

Rectangle r=new Rectangle();

r.area(2,5);

Circle c=new Circle();

c.area(5,5);

Triangle t=new Triangle();

t.area(2,5);

} }

**OUTPUT:**

area of rectangle:10.0 area of circle:78.5 area of triangle:5.0

# WEEK8

**AIM:**

**Suppose that a table named Table.txt is stored in a text file. The first line in the file is the header, and the remaining lines correspond to rows in the table. The elements are separated by commas. Write a java program to display the table using Labels in Grid Layout.**

**PROGRAM:**

**import** java.io.\*;

**import** java.util.\*;

**import** java.awt.\*;

**import** javax.swing.\*;

**class** A **extends** JFrame

{

**public** A()

{

setSize(400,400);

setDefaultCloseOperation(JFrame.*EXIT\_ON\_ CLOSE*);

GridLayout g = **new** GridLayout(0,3);

setLayout( g);

**try**

{

FileInputStream fin = **new** FileInputStream("D:/emp.txt");

Scanner sc = **new** Scanner(fin).useDelimiter(",");

String[] arrayList;

String a;

while(sc.hasNextLine())

{

a= sc.nextLine();

arrayList = a.split(",");

**for** (String i : arrayList)

{

add(**new** JLabel(i));

}

}

}

**catch** (Exception ex) {

}

*setDefaultLookAndFeelDecorated*(**true**);

pack();

setVisible(**true**);

}

}

**public class** Tbl

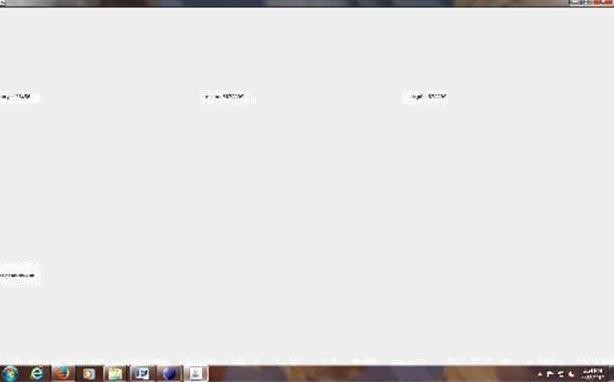
{

**public static void** main(String[] args)

{

A a = **new** A();}}

OUTPUT:



# WEEK9

**AIM:**

**Write a java program that handles all mouse events and shows the event name at the center of the window when a mouse event is fired (use Adapter classes).**

**PROGRAM**:

import javax.swing.event.\*;

import java.awt.event.\*;

import java.awt.\*;

import javax.swing.JApplet;

public class mouseevnts extends JApplet implements MouseListener

{

private int x,y;

private String event;

public void init()

{

setLayout(new FlowLayout());

x=-1;

addMouseListener(this);

}

public void paint(Graphics g)

{

super.paint(g);

g.drawRect(0,0,getWidth(),getHeight());

if(x!=1)

{

g.drawString("Mouseevent is"+event+"("+x+","+y+")", 10,50);

}

}

public void mousePressed(MouseEvent e)

{

x=e.getX();

y=e.getY();

event="pressed";

repaint();

}

public void mouseClicked(MouseEvent e)

{

x=e.getX();

y=e.getY();

event="clicked";

repaint();

}

public void mouseReleased(MouseEvent e)

{

x=e.getX();

y=e.getY(); event="Reeleased";

repaint();

}

public void mouseExited(MouseEvent e)

{

x=e.getX();

y=e.getY(); event="Exited"; repaint();

}

public void mouseEntered(MouseEvent e)

{

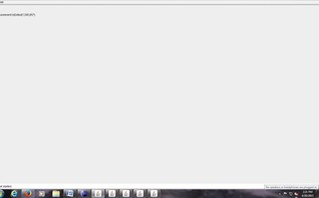
}

}

x=e.getX();

y=e.getY(); event="Entered"; repaint();

**OUTPUT:**







# WEEK10

**AIM:**

**Write a java Program that loads names and phone numbers from a text file where the data is organized as one line per record and each filed in a record are separated by a tab (\t). It takes a name or phone number as input and prints the corresponding other value from the hash table (hint: use hash tables).**

**PROGRAM:**

import java.io.\*;

import java.util.\*;

public class Phonebook

{

public static void main(String args[])

{

Try

{

FileInputStream fis=new FileInputStream("//home/gcet/Desktop/myfile.txt");

Scanner sc=new Scanner(fis).useDelimiter("\t");

Hashtable<String,String> ht=new Hashtable<String,String> ();

String[] strarray;

String a,str;

while(sc.hasNex t())

{

a=sc.nextLine();

strarray=a.split("\t");

ht.put(strarray[0],strarray[1]);

System.out.println("hash table values are"+strarray[0]+":"+strarray[1]);

}

Scanner s=new Scanner(System.in);

System.out.println("enter the name as given in the phone book");

str=s.next();

if(ht.containsKey(str))

{

System.out.println("phone no is"+ht.get(str));

}

else

{

System.out.println("name is not matched");

}

}

catch(Exception e)

{

System.out.println(e);

}}}

## Myfile.txt

Surya 1234567

Ravi 456789

Sudha 6789900

# OUTPUT:

Surya:1234567 Ravi:456789

Sudha:6789900

enter the name as given in the phone book Ravi

phone no is: 456789

enter the name as given in the phone book soni

name is not matched

# WEEK11

**AIM:**

**Write a java Program that loads names and phone numbers from a text file where the data is organized as one line per record and each filed in a record are separated by a tab (\t). It takes a name or phone number as input and prints the corresponding other value from the hash table with database instead of a text file.**

**PROGRAM:**

import java.sql.\*;

public class JdbcExample

{

static final String JDBC\_DRIVER="com.mysql.jdbc.Driver";

static final String DB\_URL="jdbc:mysql://localhost/jdbc";

static final String USER="root";

static final String PASS="Gcet@05";

public static void main(String args[])

{

Connection conn=null;

Statement stmt=null; try

{

System.out.println("connecting to database--");

conn=DriverManager.getConnection(DB\_URL,USE R,PASS);

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

stmt=conn.createStatement();

String sql;

System.out.println(“Enter the name as in database”);

String nm=sc.next();

sql="SELECT phone FROM phonenub where name="+nm;

ResultSet rs=stmt.executeQuery(sql);

while(rs.next())

{

int phone=rs.getInt("phone");

System.out.println("phone:"+phone);

}

rs.close();

stmt.close();

conn.close();

}

catch(SQLException se)

{

se.printStackTrace();

}}

# OUTPUT :

connecting to database— creating statement—

Enter the name as in database “surya”

Phone: 1234567

# 

# WEEK12

**AIM:**

**Write a java Program that takes tab separated data (one record per line) from a text file and inserts them into a database.**

**PROGRAM:**

**import** java.sql.\*;

**import** java.io.\*;

**import** java.util.\*;

**public class** Tbltodb

{

**public static void** main(String[] args)

{

Connection cn;

Statement st;

**try**

{

cn=DriverManager.*getConnection*("jdbc:mysql://localhost/jdbc","root ","Gcet@05"); st=cn.createStatement();

String sql="";

FileInputStream fin=**new** FileInputStream("D:\\myfile.txt");

Scanner sc=**new** Scanner(fin);

String[] arrayList;

String a="";

**int** i=0;

**while**(sc.hasNex t())

{

a=sc.nextLine();

arrayList=a.split("\\s+");

sql="insert into emp values("+"'"+arrayList[0]+"','"+arrayList[1]+"')";

st.execute(sql);

i++;

System.*out*.println(arrayList[0]+":"+array List[1]);

}

System.*out*.println(i+" Records are inserted");

st.close();

cn.close();

}

**catch**(Exception ex)

{

System.*out*.println(ex.getMessage());

}

}

}

Myfile.txt

Surya 1234567

Ravi 456789

Sudha 6789900

# MYSQL

MYSQL>create database jdbc; MYSQL> use jdbc;

MYSQL>create table emp(name varchar(20),phonenum int);

# OUTPUT:

Surya:1234567 Ravi:456789

Sudha:6789900

# WEEK13

**AIM**

**Write a java program that prints the meta-data of a given table.**

**PROGRAM:**

**import** java.sql.\*;

**import** java.util.\*;

**public class** Tblmdata

{

**public static void** main(String[] args)

{

Connection cn;

Statement st;

ResultSet rs, rs1;

ResultSetMetaData rsmd;

try

**{**

Scanner sc = **new** Scanner(System.*in*);

System.*out*.println("------connecting database-----");

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

String dbname = sc.next(); System.*out*.println("Enter Password");

String pass = sc.next();

cn = DriverManager.*getConnection*("jdbc:mysql://localhost/" + dbname, "root", pass);

st = cn.createStatement();

DatabaseMetaData dm = cn.getMetaData();

rs = dm.getTables(cn.getCatalog(), "%", "%", **null**);

String s = "";

String sql = "select \* from ", sql1 = ""; System.*out*.println("-----------Database is " + dbname); System.*out*.println("-------------------------");

System.*out*.println("Tables are");

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

**while** (rs.next()) { sql1 = "";

System.*out*.println("-------Table Name: " + rs.getString(3) + "---------");

sql1=sql+ rs.getString(3);

rs1= st.executeQuery(sql1);

rsmd=rs1.getMetaData();

System.*out*.println("Columns are ");

System.*out*.println("Column Name\tColumn Type\tSize");

**for** (**int** i = 1; i <= rsmd.getColumnCount(); i++)

{

System.*out*.println(rsmd.getColumnLabel(i) + "\t" + rsmd.getColumnTypeName(i) + "\t"

+rsmd.getColumnDisplaySize(i));

}

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

}

rs.close();

cn.close();

} **catch** (Exception ex) { System.*out*.println(ex.getMess age());

}

}}

|  |  |  |
| --- | --- | --- |
| **OUTPUT:** |  |  |
| ------connecting database----  - | | |
| Enter Database  Name | | |
| Jdbc |  |  |
| Enter Password | | |
| Gcet@05 |  |  |
| -----------Database is jdbc | | |
| ----------------------  --- | | |
| Tables are |  |  |
| ----------------------  --- | | |
| -------Table Name: emp-------  -- | | |
| Columns are | | |
| Column Name Column Type Size | | |
| ---------------------------------- | | |
| Id | Int | 30 |
| Name | varchar | 20 |
|  |  |  |