Contents

(This was made in 2019 by Madhav Sirohi)

1.Network Configuration and Open Source Software used in School.

Network Topology:

Bus Topology is used in the school in which there is a main powerful Server to which multiple Clients are connected. This is accomplished by using Client and Server Software.

Open Source Software Packages used:

- VLC Media Player
- My SQL
- Mozilla Firefox… etc.

2.Java Applications:

Contents

Problem1: Displaying a message using Label:

Solution:

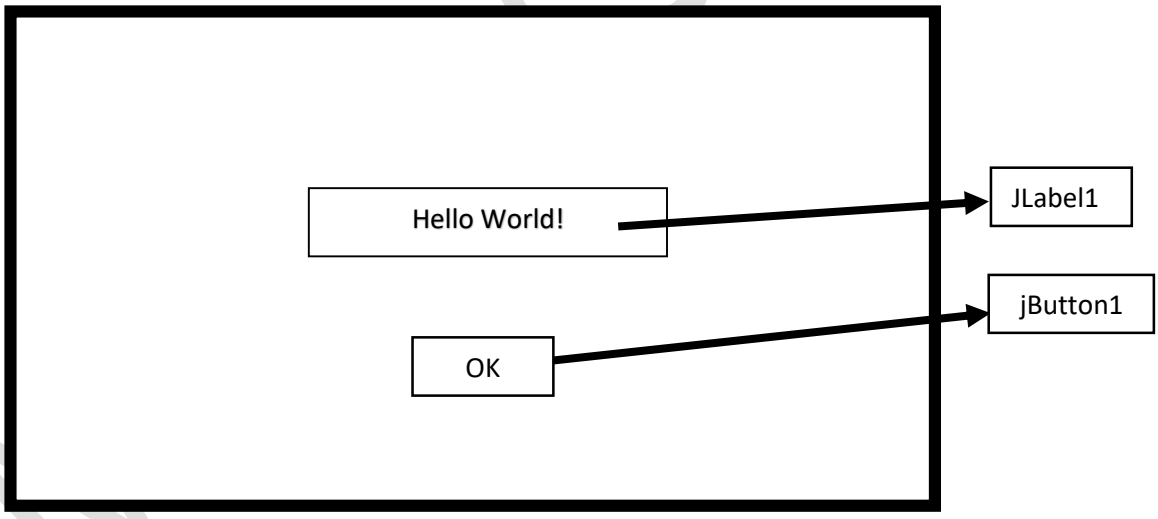
Step1: Create a jFrameForm then add a jLabel and jButton to it to it.

Step2: Set the text properties of the button to “OK” and of jLabel to null.

Step3: Double Click on the Button and add the following code in the Action Event Handler:

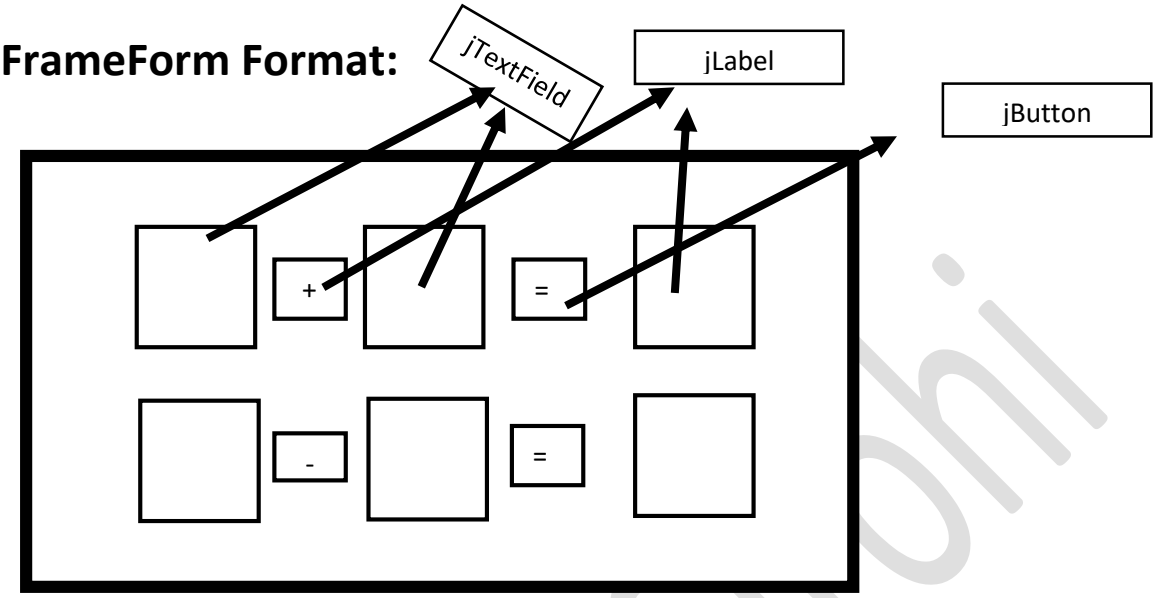
jLabel1.setText(“Hello World!”);

Output:



Problem2: Making a calculator for + and - operations:

jFrameForm Format:



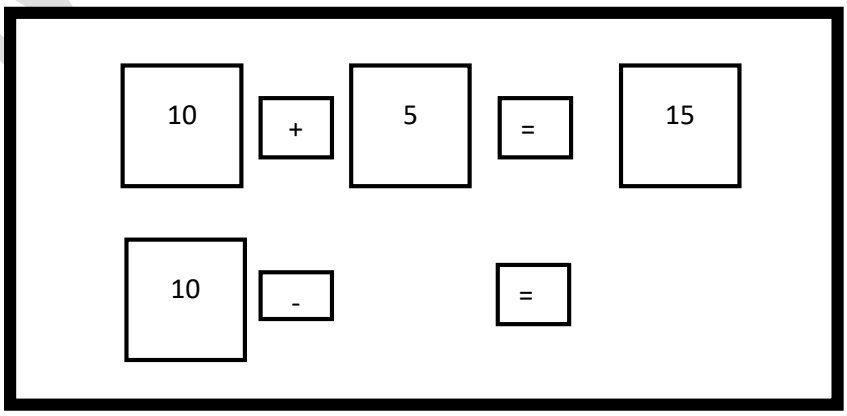
Solution:

Step1: Create the above shown jFrameForm then add jLabel(s), jTextField(s) and jButton(s) to it.

Step2: Double Click on the jButton and add the following code in the Action Event Handler:

```
int a= Integer.parseInt(jTextField1.getText());

int b=Integer.parseInt(jTextField2.getText());

jLabelSum.setText(“” + (a+b));
```

Output:



Problem3: Displaying a message using Message Dialog:

Solution:

Step1: Create a jFrameForm then add a jButton to it to it.

Step2: Set the text properties of the button to “Give me the Message.”.

Step3: Double Click on the Button and add the following code in the Action Event Handler:

JOptionPane.showMessageDialog(null,” Hello World!”);

Note: Write this on the top of the source code editor(import for JOptionPane):

Import javax.swing.jOptionPane;

Output:

Problem4: Concatenating two Strings.

Solution:

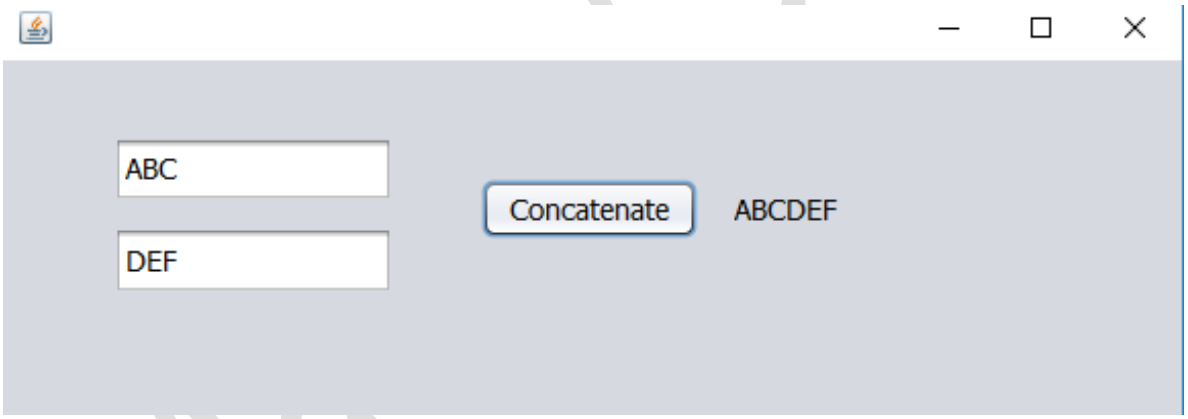
Step1: Create a jFrameForm then add a jButton, 2 jTextField(s) and a jLabel to it to it to it.

Step2: Set the text properties of the button to “Concatenate” and of the text field(s) and jTextArea to null (keep them empty).

Step3: Double Click on the Button and add the following code in the Action Event Handler:

```
jLabel1.setText(jTextField1.getText() + jTextField2.getText());
```

Output:



Problem5: Voting eligibility tester:

Solution:

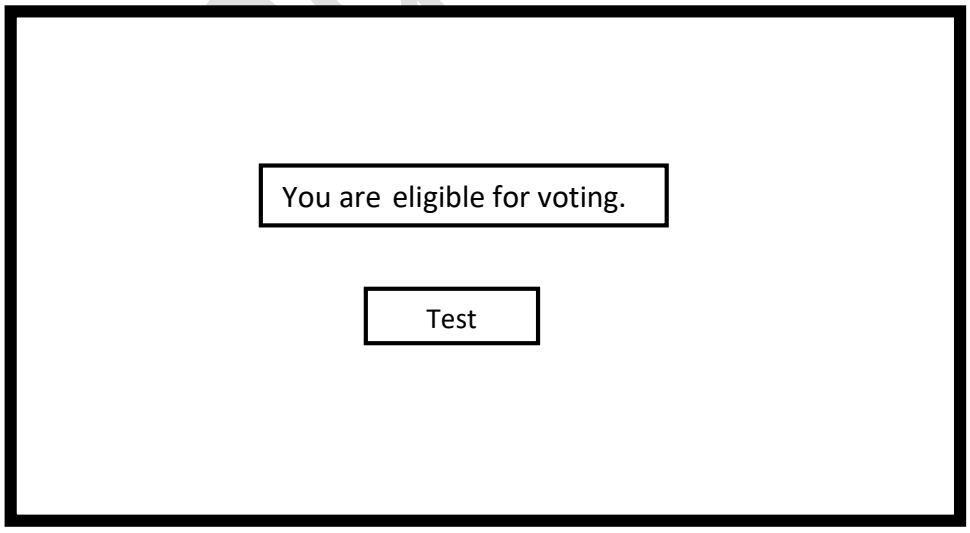
Step1: Create a jFrameForm then add a jButton and a jTextField to it to it.

Step2: Set the text properties of the button to “Test” and of the text field to null(keep it empty).

Step3: Double Click on the Button and add the following code in the Action Event Handler:

```
Int age = Integer.parseInt(jTextField1.getText());

if (age>=18)

jTextField1.setText(“You are eligible for voting.”);

else

jTextField1.setText(“You are not eligible for voting.”);
```

Output:



Problem6: Marks to Grade converter:

Solution:

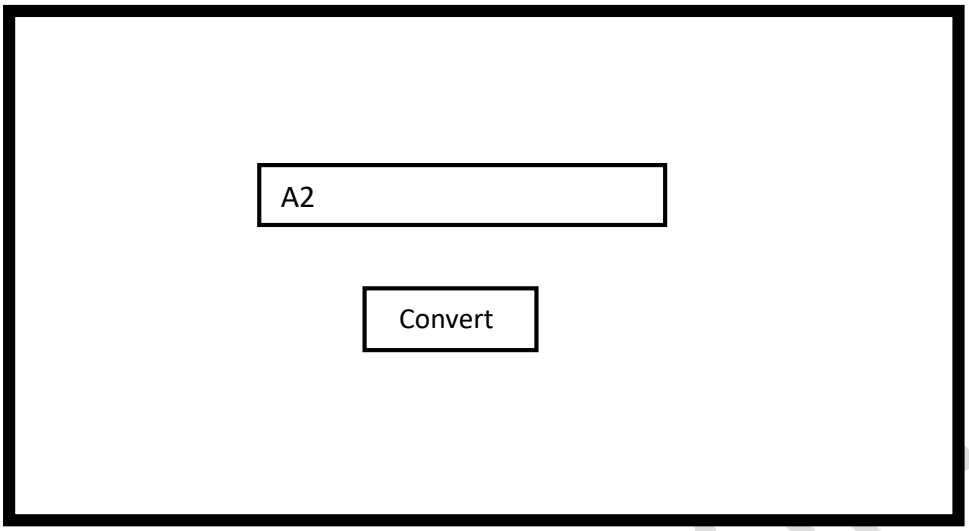
Step1: Create a jFrameForm then add a jButton and a jTextField to it to it.

Step2: Set the text properties of the button to “Covert” and of the text field to null(keep it empty).

Step3: Double Click on the Button and add the following code in the Action Event Handler:

```
Int marks = Math.round(Integer.parseInt(jTextField1.getText()));
if (marks>=91 && marks<=100)
jTextField1.setText(“A1”);
if (marks>=81 && marks<=90)
jTextField1.setText(“A2”);
if (marks>=71 && marks<=80)
jTextField1.setText(“B1”);
if (marks>=61 && marks<=70)
jTextField1.setText(“B2”);
if (marks>=51 && marks<=60)
jTextField1.setText(“C1”);
if (marks>=41 && marks<=50)
jTextField1.setText(“C2”);
if (marks>=31 && marks<=40)
jTextField1.setText(“D1”);
if (marks<=30)
jTextField1.setText(“D2”);
```

Output:



Problem7: Profit calculator:

Solution:

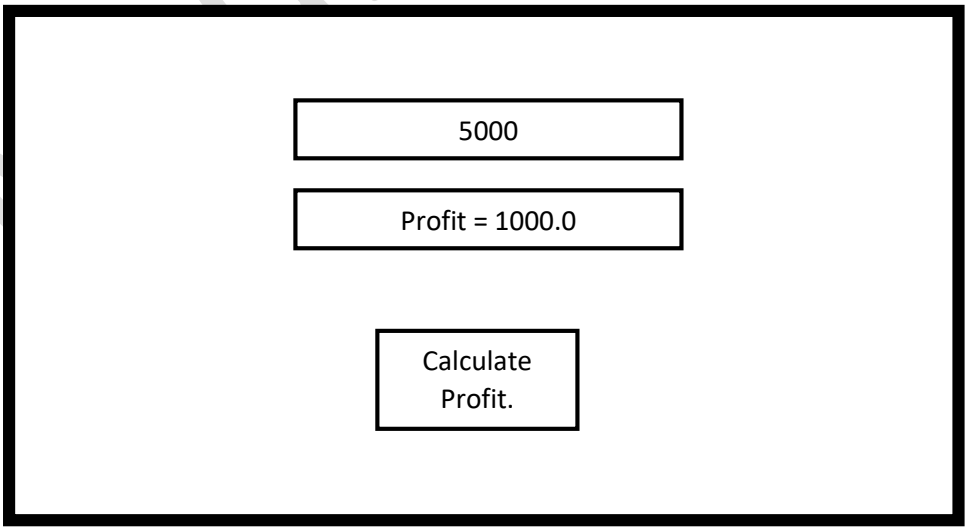
Step1: Create a jFrameForm then add a jButton and two jTextField(s) to it to it.

Step2: Set the text properties of the button to “Calculate Profit” and of the text field(s) to ‘Insert Sales here’ and null(keep it empty).

Step3: Double Click on the Button and add the following code in the Action Event Handler:

```
Double sales = Double.parseDouble(jTextField1.getText());

if (sales>=2000.00)

jTextField2.setText(“Profit =” + (sales*0.2));

else

jTextField2.setText(“No Profit”);
```

Output:



Problem8: Result calculator:

Solution:

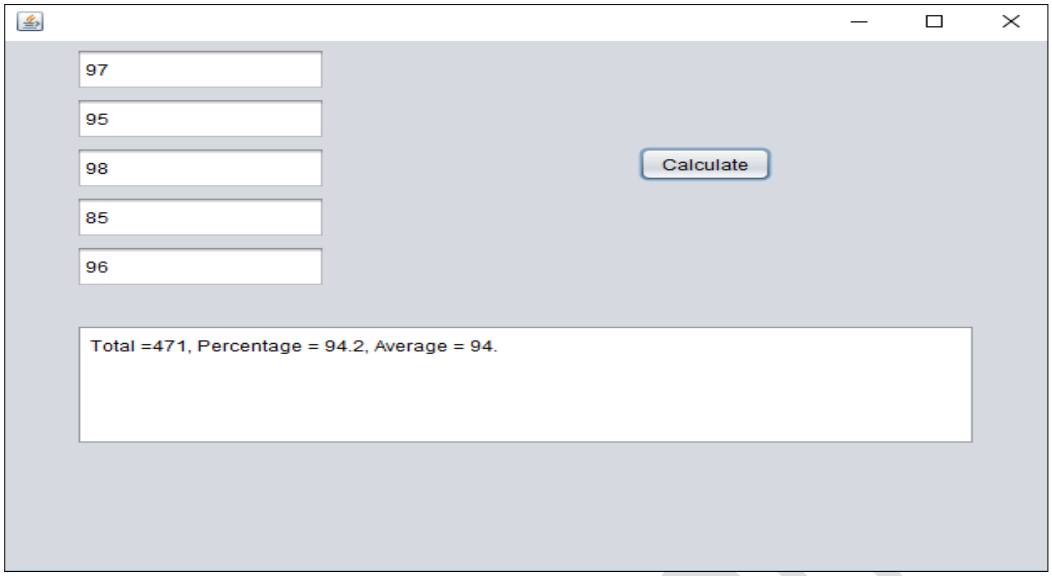
Step1: Create a jFrameForm then add a jButton, 5 jTextField(s) and a jTextArea to it to it to it.

Step2: Set the text properties of the button to “Calculate” and of the text field(s) and jTextArea to null (keep them empty).

Step3: Double Click on the Button and add the following code in the Action Event Handler:

```
int m1 = Integer.parseInt(jTextField1.getText());

int m2 = Integer.parseInt(jTextField2.getText());

int m3 = Integer.parseInt(jTextField3.getText());

int m4 = Integer.parseInt(jTextField4.getText());

int m5 = Integer.parseInt(jTextField5.getText());

int sum= m1+m2+m3+m4+m5;

jTextArea1.setText(“Total = ” + (sum) + “, ” + “Percentage = ” + (sum*0.2) + “, ”+ “Average = ” + (sum/5) + “.”);
```

Output:



Problem9: Multiplication generator:

Solution:

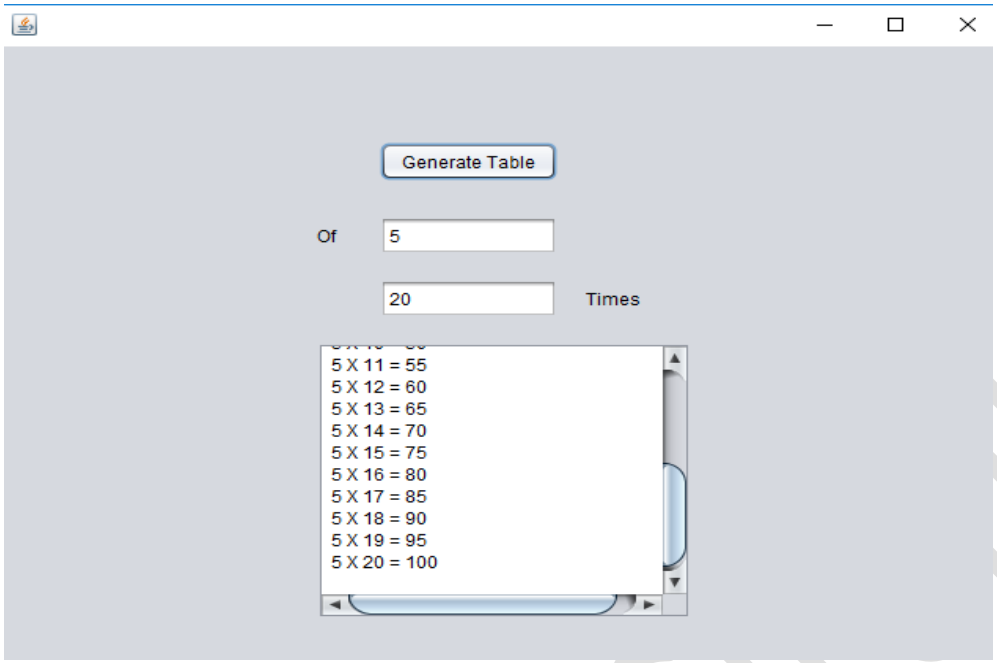
Step1: Create a jFrameForm then add a jButton,2 jTextField(s) and a jTextArea to it to it to it.

Step2: Set the text properties of the button to “Generate” and of the text field(s) and jTextArea to null (keep them empty).

Step3: Double Click on the Button and add the following code in the Action Event Handler:

```
long n=Long.parseLong(jTextField1.getText()),
t=Long.parseLong(jTextField2.getText());
   for(int i=1;i<=t;++i)
     {
        jTextArea1.append(n +" X "+ i + " = " +(i*n)+"\n");
     }
```

Output:



Problem10: Super-Market Discount Calculator:

Solution:

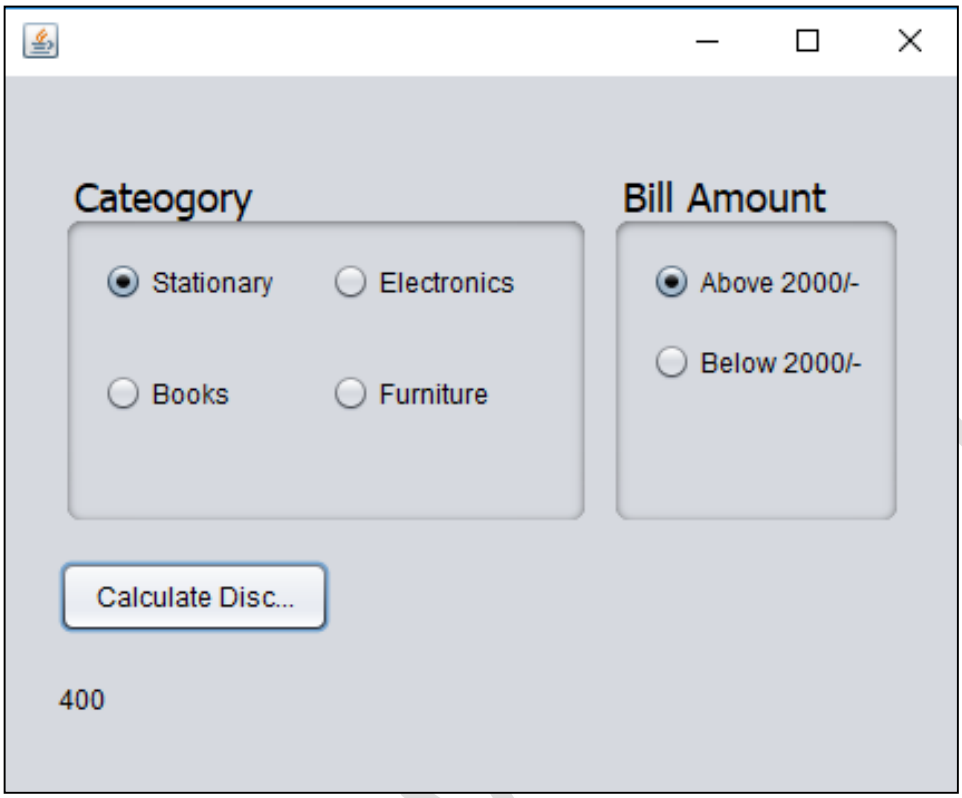
Step1: Create a jFrameForm then add a jButton, a jLabel, 2 Button-Group(s), 2 jPanel(s) and 6 jRadioButton(s) (Four in one panel and one group and the other two in the other panel and group.) to it.

Step2: Set the text properties of the button to “Calculate Discount” and of jRadioButton(s) to “Stationary”, “Electronics”, “Books”, “Furniture”, “Above 2000/-” and “Below 2000/-”.

Step3: Double Click on the Button and add the following code in the Action Event Handler:

```
int discount=0;
    if(jRadioButton1.isSelected())
      discount=500;
```

if(jRadioButton2.isSelected())



discount=300;

if(jRadioButton3.isSelected())

discount=200;

if(jRadioButton4.isSelected())

discount=100;

if(jRadioButton5.isSelected())

discount +=100;

jLabel1.setText(""+ discount);

Output:

Problem11: Changing Background Colour of a label on the basis of selected value from a list.

Solution:

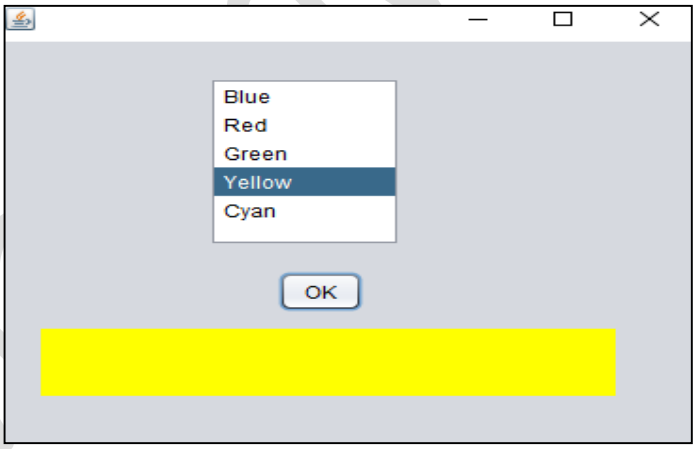
Step1: Create a jFrameForm then add a jButton, a jLabel, and a jList to it to it.

Step2: Set the text properties of the button to “OK” and of Label to null(keep it empty) and List’s model to “Blue, Red, Green, Yellow, Cyan”.

Step3: Double Click on the Button and add the following code in the Action Event Handler:

```
switch (jList1.getSelectedIndex())
   {case -1: jLabel1.setBackground(Color.white); break;
    case 0: jLabel1.setBackground(Color.blue); break;
    case 1: jLabel1.setBackground(Color.red); break;
    case 2: jLabel1.setBackground(Color.green); break;
    case 3: jLabel1.setBackground(Color.yellow); break;
    case 4: jLabel1.setBackground(Color.cyan); break; }
```

Output:



Problem12: Changing Background image of a label on the basis of selected value from a list.

Solution:

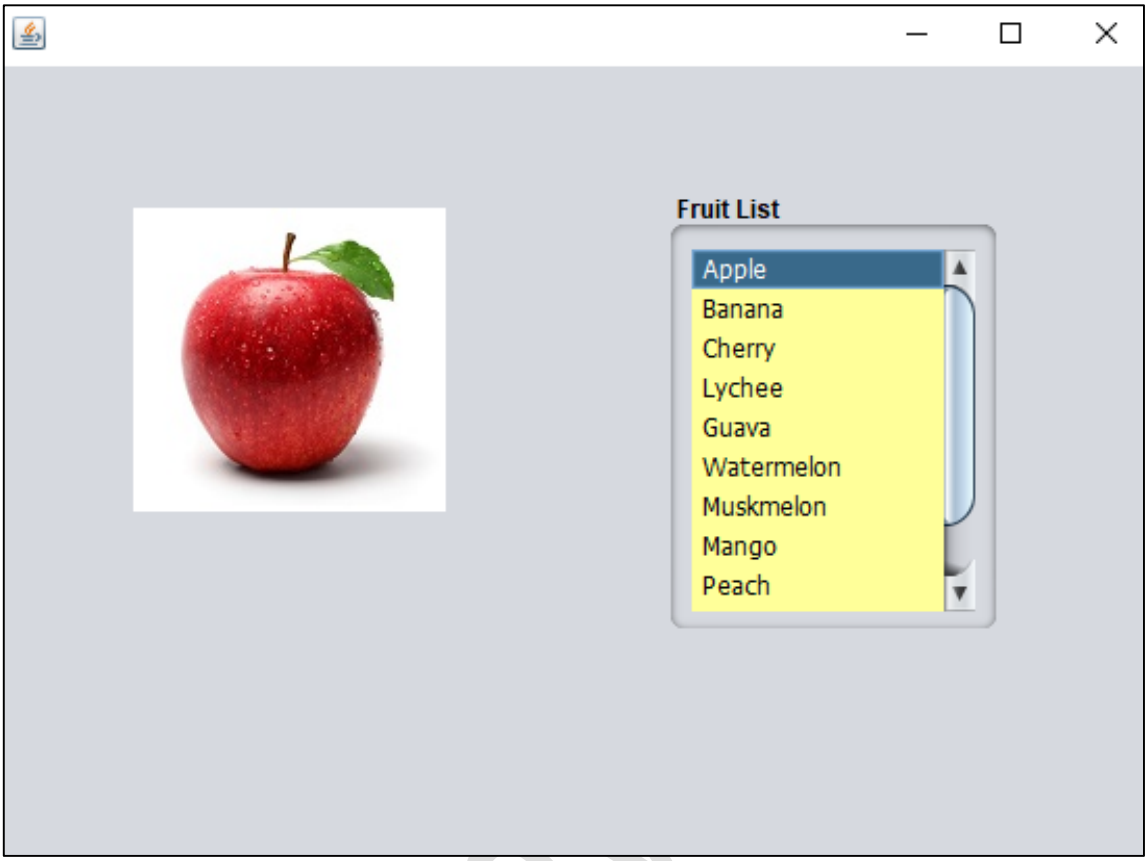
Step1: Create a jFrameForm then add a jButton, a jLabel, and a jList to it to it.

Step2: Set the text properties of the button to “OK” and of Label to null (keep it empty) and List’s model to “Apple, Banana, …”.

Step3: Add the following code in the ValueChanged Event Handler of list:

```
int i= jList1.getSelectedIndex();
    switch (i)
    {   case 0:imgLabel.setIcon(new ImageIcon("D:\\java projects\\TestGrounds\\apple.jpg")); break;
        case 1:imgLabel.setIcon(new ImageIcon("D:\\java projects\\TestGrounds\\banan.jpg"));break;
        case 2:imgLabel.setIcon(new ImageIcon("D:\\java projects\\TestGrounds\\cherr.jpg"));break;
        case 3:imgLabel.setIcon(new ImageIcon("D:\\java projects\\TestGrounds\\lichhi.jpg"));break;
        case 4:imgLabel.setIcon(new ImageIcon("D:\\java projects\\TestGrounds\\guava.jpg"));break;
        case 5:imgLabel.setIcon(new ImageIcon("D:\\java projects\\TestGrounds\\Water.jpg"));break;
        case 6:imgLabel.setIcon(new ImageIcon("D:\\java projects\\TestGrounds\\musk.jpg"));break;
        case 7:imgLabel.setIcon(new ImageIcon("D:\\java projects\\TestGrounds\\mang.jpg"));break;
        case 8:imgLabel.setIcon(new ImageIcon("D:\\java projects\\TestGrounds\\peach.jpg"));break;
        case 9:imgLabel.setIcon(new ImageIcon("D:\\java projects\\TestGrounds\\kiwi.jpg"));break;
        case 10:imgLabel.setIcon(new ImageIcon("D:\\java projects\\TestGrounds\\grap.jpg"));break;
    }
```

Output:



Problem 13: Retrieving data from dept table from test MySQL Database from within Java GUI application.

Solution:

Step1: Start NetBeans Java Application project, add a JFrame Form to it.

Step2: Add a table (namely depTbl) to the frame's design space. Edit its model property to set its number of rows as 0, Columns as 3 and Column Titles as DeptNo, Dname, Location.

Step3: Now add a button to the design space below the table and name it as rtrBtn.

Step4: Now open the Projects Window. Expand the Libraries node of your project. Right click on the Libraries node and click on Add JAR/Folder option. Then select mysql-connector-java bin file after browsing your PC and click Open, so that it gets added to the library node of your project.

Step5: In the source code editor of your project, type the following code line at the top line of it:

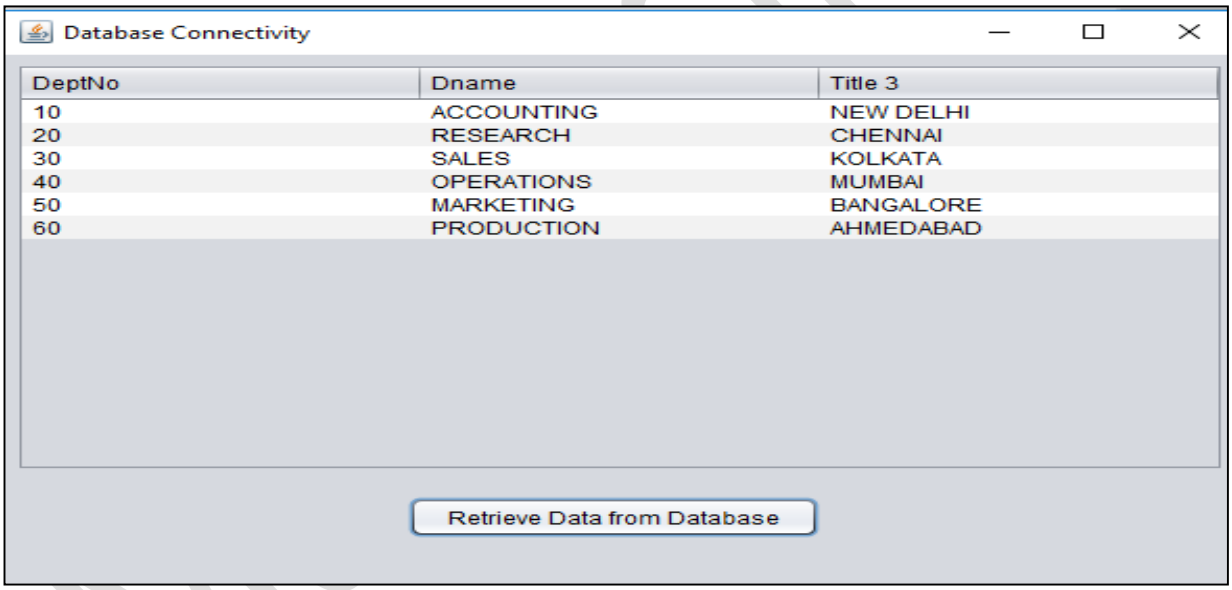
import java.sql.*;

Step6: Double Click on the Button and add the following code in the Action Event Handler:


```
DefaultTableModel model=(DefaultTableModel)depTbl.getModel();
 try {Class.forName("java.sql.Driver");
    Connection con= DriverManager.getConnection(
   "jdbc:mysql://localhost/test","root","********");
    Statement stmt=con.createStatement();
    String query="select * from dept;"; //Required Query
    ResultSet rs=stmt.executeQuery(query);
```

```
    while(rs.next())
       {String dno=rs.getString(1);
       String dname=rs.getString(2);
       String lc= rs.getString(3);
       model.addRow(new Object[]{dno,dname,lc});}
    rs.close();
    stmt.close();
    con.close();}
 catch(Exception e){ JOptionPane.showMessageDialog(null,"Error in
   Connectivity.");  }
```

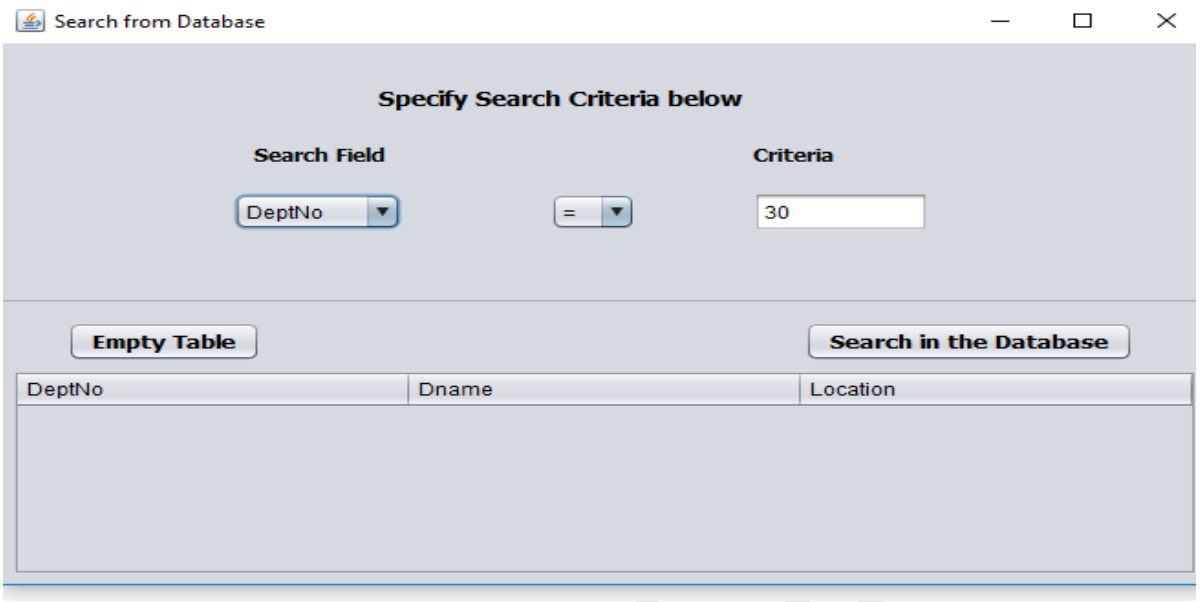
Output:



Problem 14: Retrieving data from dept table from test MySQL Database from within Java GUI application on the basis of user defined criteria.

Solution:

Step1: Start NetBeans Java Application project, add a JFrame Form to it and design it as follows:



Step2: Import required Classes and Libraries:

Import java.sql.*;

Import javax.swing.table.DefaultTableModel;

Import javax.swing.JOptionPane;

and mysql-java-connector's bin file.

Step3: Double Click on the Empty Table Button and add the following code in the Action Event Handler:

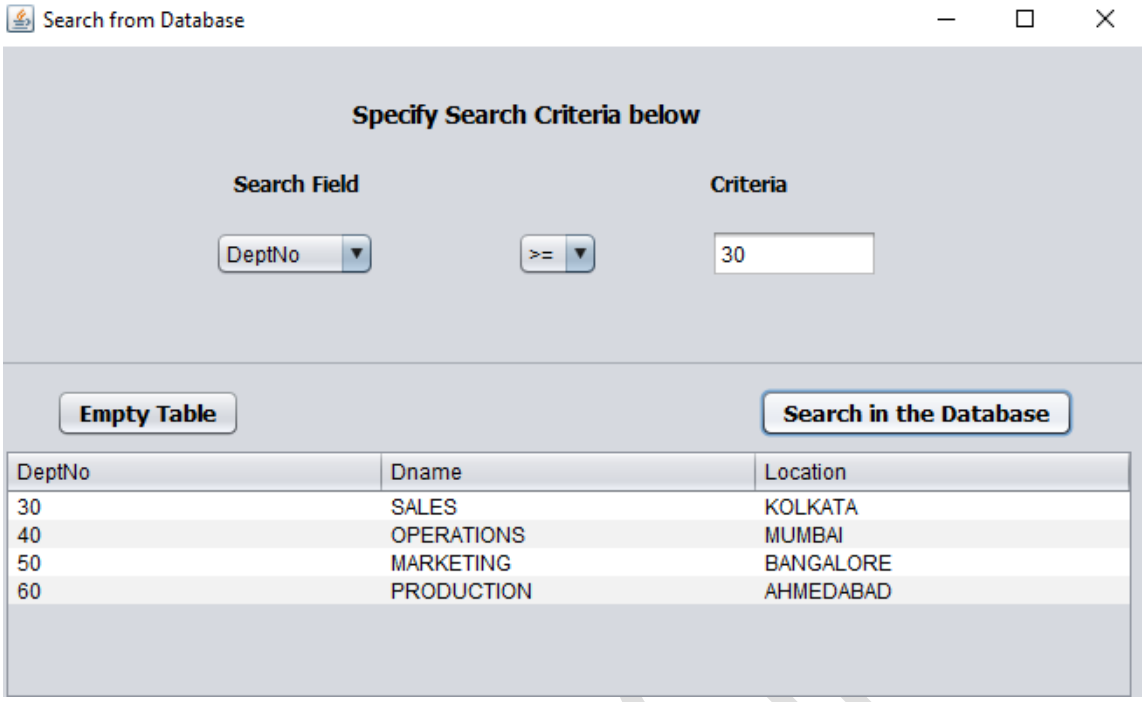
```
DefaultTableModel model=(DefaultTableModel)deptTbl.getModel();
int rows=model.getRowCount();
if(rows>0) {  for(int i=0;i<rows;i++) { model.removeRow(0); }  }
```

Step4: Double Click on the Search in Database Button and add the following code in the Action Event Handler:

```
emptyBtn.doClick();
DefaultTableModel model=(DefaultTableModel)deptTbl.getModel();
try {
```

```
Class.forName("java.sql.Driver");
Connection con=
DriverManager.getConnection("jdbc:mysql://localhost/test","root","*****")
;
Statement stmt=con.createStatement();
String sfld=(String)srchFldCBX.getSelectedItem();
String op=(String) opCbx.getSelectedItem();
String crit=criteriaTF.getText();
String query="select * from dept "+"where "+sfld+op+crit+";"; //RequiredQuery
ResultSet rs=stmt.executeQuery(query);
while(rs.next())
{       String dno=rs.getString(1);
        String dname=rs.getString(2);
        String lc= rs.getString(3);
        model.addRow(new Object[]{dno,dname,lc});      }
        rs.close();
        stmt.close();
        con.close(); }
catch(Exception e)
{
JOptionPane.showMessageDialog(null,"Error in Connectivity."); }
```

Output:



3.SQL Queries:

Contents

Query 22: Displaying current Date and Time

Query 23: Declaring Primary Key

Query 24: Dropping a Table

Query 1: Creating a Database:

Source Code: create database Project1;

Output: Query OK, 1 row affected (1.08 sec)

Query 2: Creating Table empl with columns (empno, ename, job, mgr, hiredate, sal, comm, deptno):

Source Code:

```
use Project1;

create table empl(empno int(4),ename varchar(30),job
varchar(20),mgr int(4),hiredate date,sal int(8),comm
int(8),deptno int(2));
```

Output: Database changed

Query OK, 0 rows affected (3.16 sec)

Query 3: Use desc/Describe to view the structure of a table:

Source Code: desc empl; OR describe empl:

Output:

```
mysql> describe empl;
+----------+-------------+------+-----+---------+-------+
| Field    | Type        | Null | Key | Default | Extra |
+----------+-------------+------+-----+---------+-------+
| empno    | int(4)      | YES  |     | NULL    |       |
| ename    | varchar(30) | YES  |     | NULL    |       |
| job      | varchar(20) | YES  |     | NULL    |       |
| mgr      | int(4)      | YES  |     | NULL    |       |
| hiredate | date        | YES  |     | NULL    |       |
| sal      | int(8)      | YES  |     | NULL    |       |
| comm     | int(8)      | YES  |     | NULL    |       |
| deptno   | int(2)      | YES  |     | NULL    |       |
+----------+-------------+------+-----+---------+-------+
8 rows in set (0.82 sec)
```

Query 4: Inserting a Record into the Table:

Source Code: insert into empl values(8369,'Smith','Clerk',8902,"19901218",800.00,NULL,20);

Output: Query OK, 1 row affected (0.23 sec)

Query 5: Inserting multiple Records into the Table:

Source Code: insert into empl values(8499,'Anya','Salesman',8698,"1991-2-20",1600.00,300.00,30),(8521,'Seth','Salesman',8698,"1991-2-22",1250.00,500.00,30),(8566,'Mahadevan','Manager',8839,"1991-4-2",2985.00,NULL,20),(8654,'Momin','Salesman',8698,"1991-9-28",1250.00,1400.00,30);

Output: Query OK, 4 rows affected (0.38 sec)

Records: 4 Duplicates: 0 Warnings: 0

Query 6: Creating a view from a Table

Source Code: select * from empl;

Output:

empno	ename	job	mgr	hiredate	sal	comm	deptno
8369	Smith	Clerk	8902	1990-12-18	800	NULL	20
8499	Anya	Salesman	8698	1991-02-20	1600	300	30
8521	Seth	Salesman	8698	1991-02-22	1250	500	30
8566	Mahadevan	Manager	8839	1991-04-02	2985	NULL	20
8654	Momin	Salesman	8698	1991-09-28	1250	1400	30

5 rows in set (0.00 sec)

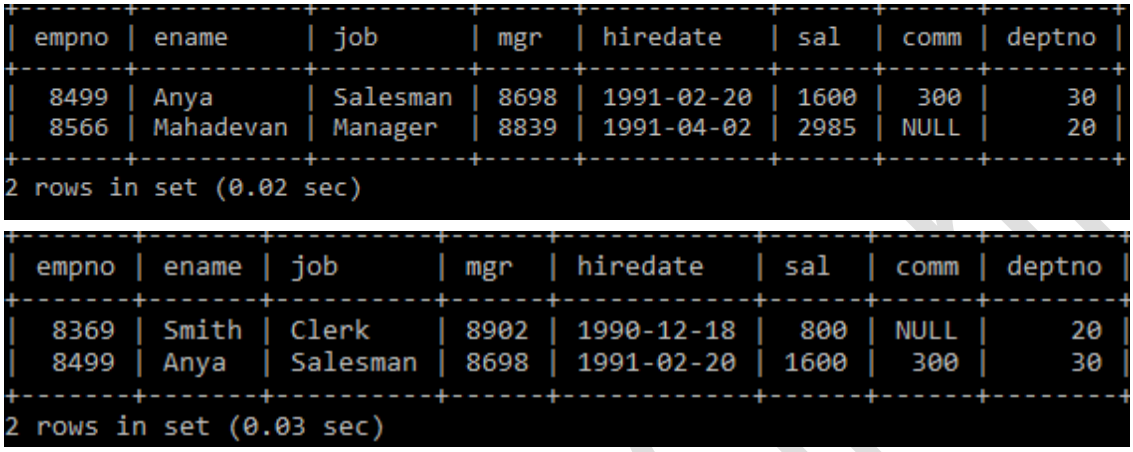
Query 7: Creating a view from a Table based on certain criteria:

Source Code:

select * from empl where sal>=1500;

select * from empl where empno<=8500;

Output:



```
+-------+-----------+----------+------+------------+------+------+--------+
| empno | ename     | job      | mgr  | hiredate   | sal  | comm | deptno |
+-------+-----------+----------+------+------------+------+------+--------+
|  8499 | Anya      | Salesman | 8698 | 1991-02-20 | 1600 |  300 |     30 |
|  8566 | Mahadevan | Manager  | 8839 | 1991-04-02 | 2985 | NULL |     20 |
+-------+-----------+----------+------+------------+------+------+--------+
2 rows in set (0.02 sec)
```

```
+-------+-------+----------+------+------------+------+------+--------+
| empno | ename | job      | mgr  | hiredate   | sal  | comm | deptno |
+-------+-------+----------+------+------------+------+------+--------+
|  8369 | Smith | Clerk    | 8902 | 1990-12-18 |  800 | NULL |     20 |
|  8499 | Anya  | Salesman | 8698 | 1991-02-20 | 1600 |  300 |     30 |
+-------+-------+----------+------+------------+------+------+--------+
2 rows in set (0.03 sec)
```


Query 8: Doing operations on Columns:

Source Code: select empno+1,ename,job from empl;

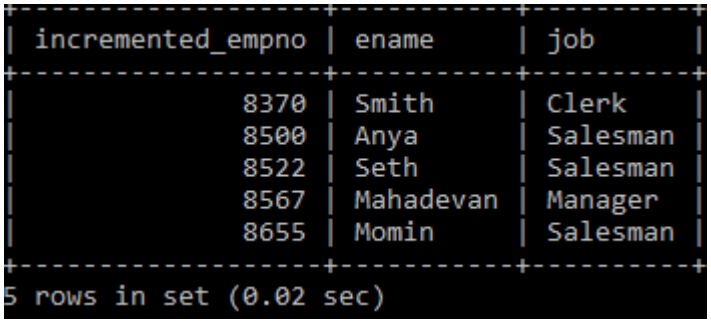
Output:

```
+---------+-----------+----------+
| empno+1 | ename     | job      |
+---------+-----------+----------+
|    8370 | Smith     | Clerk    |
|    8500 | Anya      | Salesman |
|    8522 | Seth      | Salesman |
|    8567 | Mahadevan | Manager  |
|    8655 | Momin     | Salesman |
+---------+-----------+----------+
5 rows in set (0.05 sec)
```


Query 9: Assigning Column Alias:

Source Code: select empno+1 as “incremented _empno”,ename,job from empl;

Output:



```
+-------------------+-----------+-----------+
| incremented_empno | ename     | job       |
+-------------------+-----------+-----------+
|              8370 | Smith     | Clerk     |
|              8500 | Anya      | Salesman  |
|              8522 | Seth      | Salesman  |
|              8567 | Mahadevan | Manager   |
|              8655 | Momin     | Salesman  |
+-------------------+-----------+-----------+
5 rows in set (0.02 sec)
```


Query 10: Inserting Comments:

Source Code:

```
mysql> select * from dept -- single line comment
    -> #Another single line comment
    -> /* Multi
   /*>    line
   /*>    comment*/
    -> ;
```

Output:

```
mysql> select * from dept -- single line comment
    -> #Another single line comment
    -> /* Multi
   /*>    line
   /*>    comment*/
    -> ;
+--------+------------+-----------+
| DeptNo | Dname      | Location  |
+--------+------------+-----------+
|     10 | ACCOUNTING | NEW DELHI |
|     20 | RESEARCH   | CHENNAI   |
|     30 | SALES      | KOLKATA   |
|     40 | OPERATIONS | MUMBAI    |
|     50 | MARKETING  | BANGALORE |
|     60 | PRODUCTION | AHMEDABAD |
+--------+------------+-----------+
6 rows in set (0.00 sec)
```


Query 11: Searching based on character sequence:

Source Code: mysql> select * from dept where Dname like 'A%';

Output:

```
mysql> select * from dept where Dname like 'A%';
+--------+------------+-----------+
| DeptNo | Dname      | Location  |
+--------+------------+-----------+
|     10 | ACCOUNTING | NEW DELHI |
+--------+------------+-----------+
1 row in set (0.15 sec)
```


Query 12: Use of ALL keyword:

Source Code: mysql> select all location from dept;

Output:

```
mysql> select all location from dept;
+-----------+
| location  |
+-----------+
| NEW DELHI |
| CHENNAI   |
| KOLKATA   |
| MUMBAI    |
| BANGALORE |
| AHMEDABAD |
+-----------+
6 rows in set (0.28 sec)
```


Query 13: Performing simple Calculations:

Source Code: mysql> select all location from dept;

Output:

```
mysql> select 12*13+1;
+---------+
| 12*13+1 |
+---------+
|     157 |
+---------+
1 row in set (0.03 sec)
```


Query 14: Range based selection:

Source Code: mysql> select * from dept where DeptNo between 20 AND 40;

Output:

```
mysql> select * from dept where DeptNo between 20 AND 40;
+--------+------------+----------+
| DeptNo | Dname      | Location |
+--------+------------+----------+
|     20 | RESEARCH   | CHENNAI  |
|     30 | SALES      | KOLKATA  |
|     40 | OPERATIONS | MUMBAI   |
+--------+------------+----------+
3 rows in set (0.00 sec)
```


Query 15: List based Selection:

Source Code:

1. **mysql> select * from dept where Location in('NEW DELHI','KOLKATA','MUMBAI');**
2. **mysql> select * from dept where Location not in('NEW DELHI','KOLKATA','MUMBAI');**

Output:

1.

```
mysql> select * from dept where Location in('NEW DELHI','KOLKATA','MUMBAI');
+--------+------------+-----------+
| DeptNo | Dname      | Location  |
+--------+------------+-----------+
|     10 | ACCOUNTING | NEW DELHI |
|     30 | SALES      | KOLKATA   |
|     40 | OPERATIONS | MUMBAI    |
+--------+------------+-----------+
3 rows in set (0.00 sec)
```

2.

```
mysql> select * from dept where Location not in('NEW DELHI','KOLKATA','MUMBAI');
+--------+------------+-----------+
| DeptNo | Dname      | Location  |
+--------+------------+-----------+
|     20 | RESEARCH   | CHENNAI   |
|     50 | MARKETING  | BANGALORE |
|     60 | PRODUCTION | AHMEDABAD |
+--------+------------+-----------+
3 rows in set (0.00 sec)
```

Query 16: Searching for NULL:

Source Code:

1. **mysql> select * from dept where Location is NULL;**
2. **mysql> select * from dept where Location is not NULL;**

Output:

1.

```
mysql> select * from dept where Location is NULL;
Empty set (0.00 sec)
```

2.

```
mysql> select * from dept where Location is not NULL;
+--------+------------+-----------+
| DeptNo | Dname      | Location  |
+--------+------------+-----------+
|     10 | ACCOUNTING | NEW DELHI |
|     20 | RESEARCH   | CHENNAI   |
|     30 | SALES      | KOLKATA   |
|     40 | OPERATIONS | MUMBAI    |
|     50 | MARKETING  | BANGALORE |
|     60 | PRODUCTION | AHMEDABAD |
+--------+------------+-----------+
6 rows in set (0.00 sec)
```


Query 17: Sorting Results of select statement:

Source Code: mysql> select * from dept order by Dname;

Output:

```
mysql> select * from dept order by Dname;
+--------+------------+-----------+
| DeptNo | Dname      | Location  |
+--------+------------+-----------+
|     10 | ACCOUNTING | NEW DELHI |
|     50 | MARKETING  | BANGALORE |
|     40 | OPERATIONS | MUMBAI    |
|     60 | PRODUCTION | AHMEDABAD |
|     20 | RESEARCH   | CHENNAI   |
|     30 | SALES      | KOLKATA   |
+--------+------------+-----------+
6 rows in set (0.12 sec)
```


Query 18: Using char() and ascii() function:

Source Code:

1. mysql> select char(65);
2. mysql> select ascii("a");

Output:

1.

```
mysql> select char(65);
+----------+
| char(65) |
+----------+
| A        |
+----------+
1 row in set (0.00 sec)
```

2.

```
mysql> select ascii("a");
+------------+
| ascii("a") |
+------------+
|         97 |
+------------+
1 row in set (0.00 sec)
```


Query 19: Using Concat() function:

Source Code: mysql> select DeptNo, concat(Dname,' is located in ',Location,'.') as "Dname with Location" from dept;

Output:

```
mysql> select DeptNo, concat(Dname,' is located in ',Location,'.') as "Dname with Location" from dept;
+--------+-------------------------------------+
| DeptNo | Dname with Location                 |
+--------+-------------------------------------+
|     10 | ACCOUNTING is located in NEW DELHI. |
|     20 | RESEARCH is located in CHENNAI.     |
|     30 | SALES is located in KOLKATA.        |
|     40 | OPERATIONS is located in MUMBAI.    |
|     50 | MARKETING is located in BANGALORE.  |
|     60 | PRODUCTION is located in AHMEDABAD. |
+--------+-------------------------------------+
6 rows in set (0.00 sec)
```


Query 20: Using Substring() function:

Source Code: mysql> select DeptNo,Dname,Location,substr(Location,1,3) as 'Location code'from dept;

Output:

```
mysql> select DeptNo,Dname,Location,substr(Location,1,3) as 'Location code'from dept;
+--------+------------+-----------+---------------+
| DeptNo | Dname      | Location  | Location code |
+--------+------------+-----------+---------------+
|     10 | ACCOUNTING | NEW DELHI | NEW           |
|     20 | RESEARCH   | CHENNAI   | CHE           |
|     30 | SALES      | KOLKATA   | KOL           |
|     40 | OPERATIONS | MUMBAI    | MUM           |
|     50 | MARKETING  | BANGALORE | BAN           |
|     60 | PRODUCTION | AHMEDABAD | AHM           |
+--------+------------+-----------+---------------+
6 rows in set (0.00 sec)
```


Query 21: Using Lower()/Lcase() function:

Source Code: mysql> select *,lower(Location) from dept;

Output:

```
mysql> select *,lower(Location) from dept;
+--------+------------+-----------+-----------------+
| DeptNo | Dname      | Location  | lower(Location) |
+--------+------------+-----------+-----------------+
|     10 | ACCOUNTING | NEW DELHI | new delhi       |
|     20 | RESEARCH   | CHENNAI   | chennai         |
|     30 | SALES      | KOLKATA   | kolkata         |
|     40 | OPERATIONS | MUMBAI    | mumbai          |
|     50 | MARKETING  | BANGALORE | bangalore       |
|     60 | PRODUCTION | AHMEDABAD | ahmedabad       |
+--------+------------+-----------+-----------------+
6 rows in set (0.06 sec)
```


Query 22: Displaying current Date and Time:

Source Code: mysql> select sleep(2),sysdate(),now();

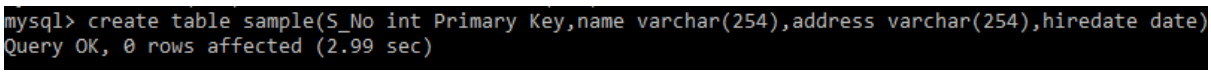
Output:

```
mysql> select sleep(2),sysdate(),now();
+----------+---------------------+---------------------+
| sleep(2) | sysdate()           | now()               |
+----------+---------------------+---------------------+
|        0 | 2019-01-09 18:56:37 | 2019-01-09 18:56:35 |
+----------+---------------------+---------------------+
1 row in set (2.00 sec)
```


Query 23: Declaring Primary Key:

Source Code: mysql> create table sample(S_No int Primary Key,name varchar(254),address varchar(254),hiredate date);

Output:



Query 24: Dropping a Table:

Source Code: mysql> drop table sample;

Output:

```
mysql> drop table sample;
Query OK, 0 rows affected (1.51 sec)
```


4.Two Web pages made with HTML

Home Page:

Source Code:

```
<HTML>

<HEAD>

        <TITLE>AnimalPedia</TITLE>

</HEAD>

<BODY topmargin=50 leftmargin=20  text=white bgcolor=teal link=pink
vlink=Black>

<font face="Amarillo" size=4 color=cream> <H1
align=center>Animalpedia</H1> </font>

<font size=4>

<P align=center><font size=5>A</font>nimal: Any of the living Organisms
belonging to the Kingdom Animalia.<BR>Eg. Sponges, Hydra, Flatworms,
Butterfly, Lemurs, Humans and many more...</P>

<P aling=left>Animals are multicellular eukaryotic organisms that form the
biological kingdom Animalia. With few exceptions, animals consume organic
material, breathe oxygen, are able to move, reproduce sexually, and grow from
a hollow sphere of cells, the blastula, during embryonic development.Over 1.5
million living animal species have been described—of which around 1 million
are insects—but it has been estimated there are over 7 million animal species
in total. Animals range in length from 8.5 millionths of a metre to 33.6 metres
(110 ft) and have complex interactions with each other and their
environments, forming intricate food webs. The study of animals is called
<I>Zoology</I>.(From <B>Wikipedia</B>)</P>

<Center>-<U>Some wonderful Animals</U>-</Center>

<HR noshade width=80% size=10 color=white>

<HR noshade width=70% size=8 color=white>

<HR noshade width=60% size=6 color=white><BR><BR>

<Center><img src=lion.jpg alt="Lion" border=5>

        <img src=koala.jpg alt="Koala" border=5>
```

```
        <img src=shark.jpg alt="Shark" border=5>
</Center>
<P>Above Shown Animals:</P>
<DL compact>
<DT>Lion:
<DD>The lion is a species in the family Felidae; it is a muscular, deep-chested cat with a short, rounded head, a reduced neck and round ears, and a hairy tuft at the end of its tail.(From Wikipedia)
<DT>Koala:
<DD>The koala is an arboreal herbivorous marsupial native to Australia. It is the only extant representative of the family Phascolarctidae and its closest living relatives are the wombats, which comprise the family Vombatidae.(From Wikipedia)
<DT>Shark:
<DD>Sharks are a group of elasmobranch fish characterized by a cartilaginous skeleton, five to seven gill slits on the sides of the head, and pectoral fins that are not fused to the head. Modern sharks are classified within the clade Selachimorpha and are the sister group to the rays.(From Wikipedia)
</DL>
<Center>Support us by filling our Form, <A href=Form_1.html title="Link to open the Form.">Here</A>.</Center>
</font><!---Basic font for the text--->
</BODY>
</HTML>
```


Form for the Home Page:

```
<HTML>
```

```
<HEAD><TITLE>Submit_Form</TITLE></HEAD>

<BODY topmargin=50 leftmargin=20>

<CENTER><H1>Form for submitting Animal
Information</H1></CENTER>

<FORM name=f1 action="Confirm.html" method="get">

Common Name: <INPUT type=text  name=nm value="type here"
maxlength=30>

<BR><BR>Scientific Name: <INPUT type=text name=s_nm>

<BR><BR>

    Eating Habits:

    <Table border=1><TR><TD  valign=top>

    <BR>

    <INPUT type="radio" name="rg" value="Herbivore">Herbivore
<INPUT type="radio" name="rg" value="Carnivore">Carnivore

    <BR>

    <INPUT type="radio" name="rg" value="Omnivore">Omnivore
<INPUT type="radio" name="rg" value="Scavengers">Scavengers

    </TD></TR><Table>

<BR><BR>

Habitat: <INPUT type=test name="hb">

<BR><BR>

Behaviour:

<BR><textarea rows=5 cols=50 name=bh>Enter Details...</textarea>

<BR><BR>

Special Identification Traits:
```


<INPUT type=checkbox name=c1 value="Contrasting Strips">Contrasting Strips <INPUT type=checkbox name=c2 value="Contrasting Spots">Contrasting Spots <INPUT type=checkbox name=c3 value="Huge Body Size">Huge Body Size <INPUT type=checkbox name=c4 value="Claws">Claws

If any other, then specify here:

<INPUT type=text name=id_t size=50>

<INPUT type=submit name=sub value="Submit Above Information">

<INPUT type=reset name=res value="Reset">

</FORM>

<A href=Animal_Information><Center>Return to Homepage</Center></As>

</BODY>

</HTML>

Confirm Page:

<HTML>

<HEAD><TITLE>CONFIRMATION_PAGE</TITLE></HEAD>

<BODY topmargin=50> <HR><HR width=80%><HR width=60%>

<H1 align=center>Form Submitted.</H1>

<HR width=60%><HR width=80%><HR>

</BODY> </HTML>