

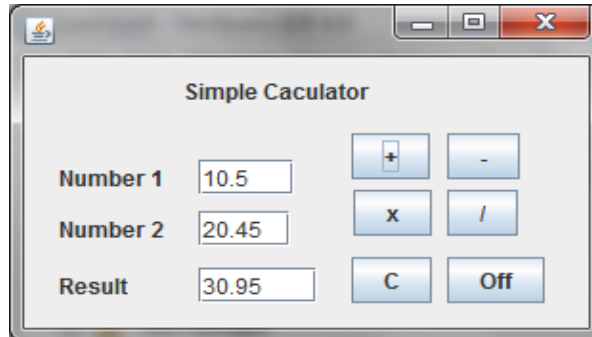
Class XII- Informatics Practices

PRACTICAL – ASSIGNMENTS

Experiment No. 1:

Objective: Understanding and use of variables of float and other data types.

Task: Develop a simple Calculator application as per given screen snapshot, to implement +, -, x and / operations. The text boxes get cleared when 'C' button is clicked.



```
Coding: private void BtnPlusActionPerformed(java.awt.event.ActionEvent evt) {
    float x,y,z;
    x=Float.parseFloat(TxtNum1.getText());
    y=Float.parseFloat(TxtNum2.getText());
    z=x+y;
    TxtResult.setText(""+z);
}

private void BtnMinusActionPerformed(java.awt.event.ActionEvent evt) {
    float x,y,z;
    x=Float.parseFloat(TxtNum1.getText());
    y=Float.parseFloat(TxtNum2.getText());
    z=x-y;
    TxtResult.setText(""+z);
}

private void BtnMulActionPerformed(java.awt.event.ActionEvent evt) {
    float x,y,z;
    x=Float.parseFloat(TxtNum1.getText());
    y=Float.parseFloat(TxtNum2.getText());
    z=x*y;
    TxtResult.setText(""+z);
}

private void BtnDivActionPerformed(java.awt.event.ActionEvent evt) {
    float x,y,z;
    x=Float.parseFloat(TxtNum1.getText());
    y=Float.parseFloat(TxtNum2.getText());
    z=x/y;
    TxtResult.setText(""+z);
}

private void BtnClearActionPerformed(java.awt.event.ActionEvent evt) {
    TxtNum1.setText("");
    TxtNum2.setText("");
    TxtResult.setText("");
}

private void BtnOffActionPerformed(java.awt.event.ActionEvent evt) {
    System.exit(0);
}
```

Experiment No. 2:

Objective: Understanding and use of Nested conditions in the Real life applications.

Task: A Quick Fox Transport Co. wants to develop an application for calculating amount based on distance and weight of goods.

The charges (Amount) to be calculated as per rates given below.

Distance	Weight	Charges per Km.
≥ 500 Km	≥ 100 kg.	Rs. 5/-
	≥ 10 and < 100 kg.	Rs. 6/-
	< 10 kg.	Rs. 7/-
< 500 Km	≥ 100 Kg.	Rs.8/-
	< 100 Kg.	Rs.5/-

Quick Fox Transport Co.

From (City)

To (City)

Weight (Kg)

Distance (Km)

Amount

```
Coding: private void BtnCalActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    float amt;  
    float wt=Float.parseFloat(TxtWeight.getText());  
    float km=Float.parseFloat(TxtDist.getText());  
    if (km >= 500)  
    {  
        if (wt >= 100)  
            amt = km*5;  
        else if (wt >= 10)  
            amt = km * 6;  
        else  
            amt = km * 7;  
    }  
    else  
    {  
        if (wt >= 100)  
            amt = km*8;  
        else  
            amt = km * 10;  
    }  
    TxtAmt.setText("" + amt);  
}
```

```
private void BtnExitActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    System.exit(0);  
}
```

Experiment No. 3:

Objective: To calculate the total marks, percentage and grades on the basis of percentage as given below.

Percentage	Grades
≥ 90	A+
≥ 80	A
≥ 70	B
≥ 60	C
≥ 50	D
≥ 40	E
< 40	F

The screenshot shows a Java Swing window titled "ABC Public School". It has a light gray background and a blue title bar. The window contains several input fields and buttons. The "Name" field contains "Manoj". The "Class" field contains "10" and the "Roll No." field contains "24". There is a section titled "Marks" with a light blue background, containing five input fields: "English" (85), "Hindi" (76), "Science" (92), "Social Sc." (78), and "Maths" (89). To the right of this section, there are three output fields: "Total Marks" (420.0), "Percentage" (84.0), and "Grade" (A). At the bottom of the window, there are two buttons: "Calculate" and "Close".

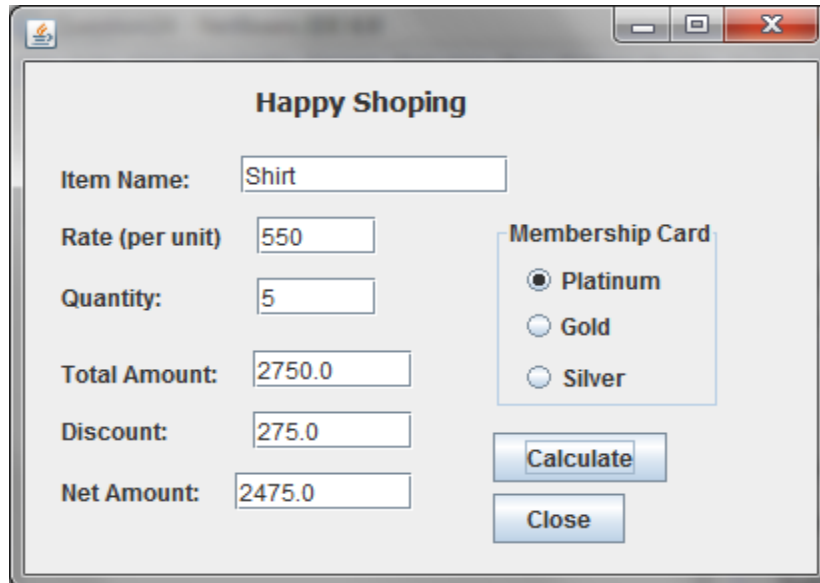
```
Coding: private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
    float eng,hin,sc,ssc,math,tm,per;  
    String gr;  
    eng=Float.parseFloat(TxtEng.getText());  
    hin=Float.parseFloat(TxtHindi.getText());  
    sc=Float.parseFloat(TxtSc.getText());  
    ssc=Float.parseFloat(TxtSSc.getText());  
    math=Float.parseFloat(TxtMaths.getText());  
    tm=eng + hin + sc + ssc + math;  
    per=tm*100/500;  
    if(per>=90)  
        gr="A+";  
    else if(per>=80)  
        gr="A";  
    else if(per>=70)  
        gr="B";  
    else if (per>=60)  
        gr="C";  
    else if (per>=50)  
        gr="D";  
    else if (per>=40)  
        gr="E";  
    else  
        gr="F";  
    TxtTMarks.setText(""+tm);  
    TxtPer.setText(""+per);  
    TxtGrade.setText(gr);  
}  
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {  
    System.exit(0);  
}
```

Experiment No. 4:

Objective: Understanding and using the Radio Button in Real-life application to determine the selection of choice and calculations accordingly.

Task: Develop a Billing application for Happy Shopping- A retail chain involved in sales of Readymade garments. The happy Shopping offers discount to its members holding Platinum, Gold and Silver card.

The 10% discount is given to Platinum card, 8% to Gold Card and 5% to Silver Card holders on sales amount.

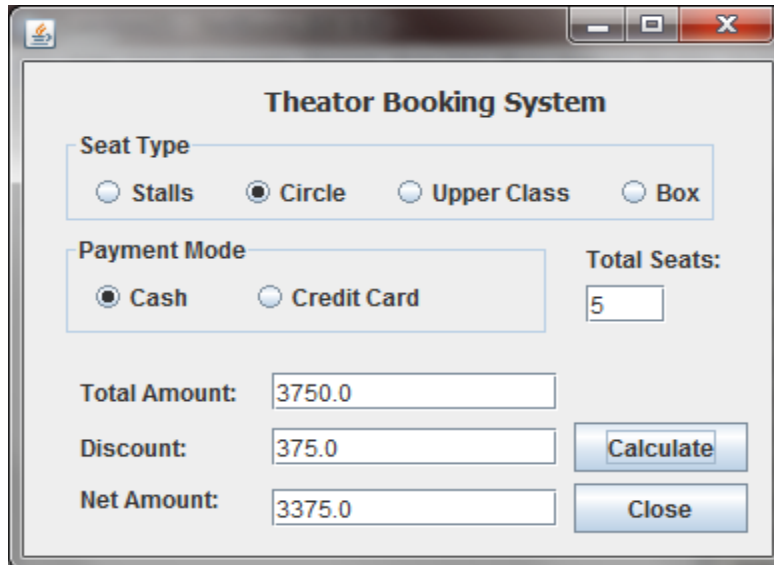


```
Coding: private void BtnCalculateActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    float rt,qty,amt,dis,net;  
    rt=Float.parseFloat(TxtRate.getText());  
    qty=Float.parseFloat(TxtQty.getText());  
    amt=rt*qty;  
    if(RdBtnPlatinum.isSelected() == true)  
        dis=amt*10/100;  
    else if(RdBtnGold.isSelected() == true)  
        dis=amt*8/100;  
    else  
        dis=amt*5/100;  
    net=amt-dis;  
    TxtAmt.setText(""+amt);  
    TxtDisc.setText(""+dis);  
    TxtNet.setText(""+net);  
}  
  
private void BtnCloseActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    System.exit(0);  
}
```

Experiment No. 5:

Objective: Understanding and using the Radio Button in Real-life application to determine the selection of choices and calculations accordingly.

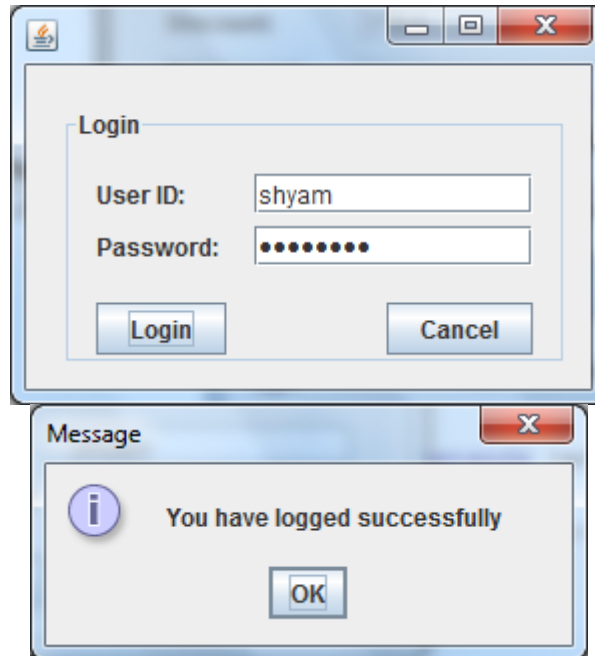
Task: The Entertainment Paradise- A theater in Delhi wants to develop a computerized Booking System. The proposed Interface is given below. The theater offers different types of seats. The Ticket rates are- Stalls- Rs. 625/-, Circle- Rs.750/-, Upper Class- Rs.850/- and Box- Rs.1000/-. A discount is given 10% of total amount if tickets are purchased on Cash. In case of credit card holders 5% discount is given.



```
Coding: private void BtnCalActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    int st;  
    float amt=0,dis=0,net;  
    st= Integer.parseInt(TxtSeat.getText());  
    if(RdBtnStall.isSelected() == true)  
        amt=st*625;  
    if(RdBtnCircle.isSelected() == true)  
        amt=st*750;  
    if(RdBtnUpper.isSelected() == true)  
        amt=st*850;  
    if(RdBtnBox.isSelected() == true)  
        amt=st*1000;  
    if(RdBtnCash.isSelected() == true)  
        dis=amt*10/100;  
    if(RdBtnCredit.isSelected() == true)  
        dis=amt*5/100;  
    net=amt-dis;  
    TxtAmt.setText(""+amt);  
    TxtDisc.setText(""+dis);  
    TxtNet.setText(""+net);  
}  
  
private void BtnCloseActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    System.exit(0);  
}
```

Experiment No. 6:

Objective: To check whether the given user id and password is correct or not.

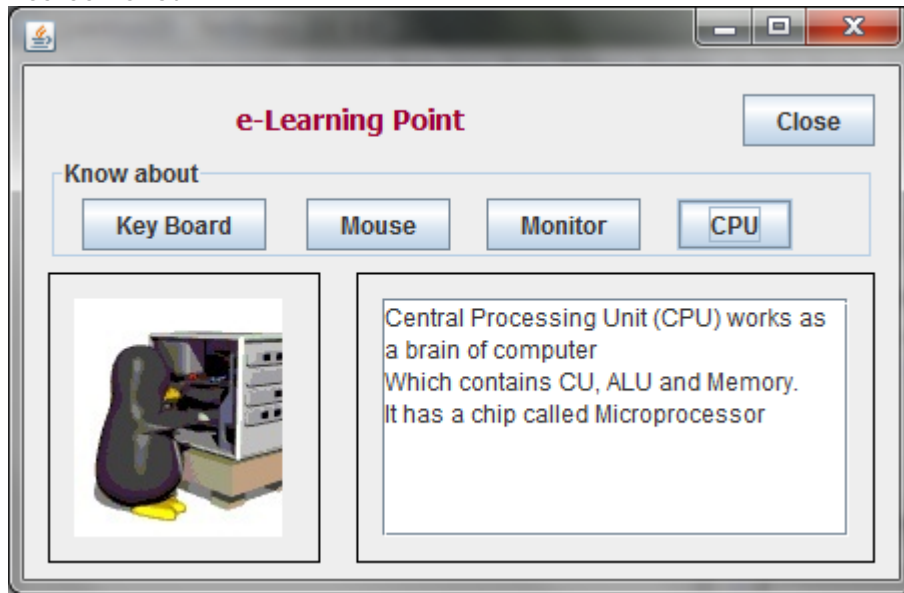


```
Coding: private void BtnLoginActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    String pwd= new String(jPasswordField1.getPassword());  
    if(pwd.equals("password"))  
        JOptionPane.showMessageDialog(null," You have logged successfully");  
    else  
        JOptionPane.showMessageDialog(null," Incorrect User ID or password");  
}  
  
private void BtnCancelActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    System.exit(0);  
}
```

Experiment No. 7:

Objective: Displaying images on a Label and Text Area control.

Task: Develop an e-Learning application with images and text information as per given screen shot.



```
Coding: private void BtnKeyboardActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    JTextArea1.setText("");
    Lblpic.setIcon(new ImageIcon("c:\\pics\\keyboard.jpg"));
    JTextArea1.append("Key Board is an input device."+"\n");
    JTextArea1.append("Which facilitates user to input text data."+"\n");
    JTextArea1.append("It has Alphabets keys, Numeric keys, Arrow keys and control keys");
}

private void BtnMouseActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    JTextArea1.setText("");
    Lblpic.setIcon(new ImageIcon("c:\\pics\\mouse.gif"));
    JTextArea1.append("Mouse is a pointing device."+"\n");
    JTextArea1.append("Which facilitates user to select something on the screen"+"\n");
    JTextArea1.append("It has two or three buttons.");
}

private void BtnMonitorActionPerformed(java.awt.event.ActionEvent evt) {
    JTextArea1.setText("");
    Lblpic.setIcon(new ImageIcon("c:\\pics\\monitor.bmp"));
    JTextArea1.append("Monitor is an softcopy output device."+"\n");
    JTextArea1.append("Which facilitates user to see text, movie on the screen"+"\n");
    JTextArea1.append("It may be TFT or CRT in different sizes.");
}

private void BtnCPUActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    JTextArea1.setText("");
    Lblpic.setIcon(new ImageIcon("c:\\pics\\cpu.jpg"));
    JTextArea1.append("Central Processing Unit (CPU) works as a brain of computer"+"\n");
    JTextArea1.append("Which contains CU, ALU and Memory."+"\n");
    JTextArea1.append("It has a chip called Microprocessor");
}

private void BtnCloseActionPerformed(java.awt.event.ActionEvent evt) {
    System.exit(0);
}
```

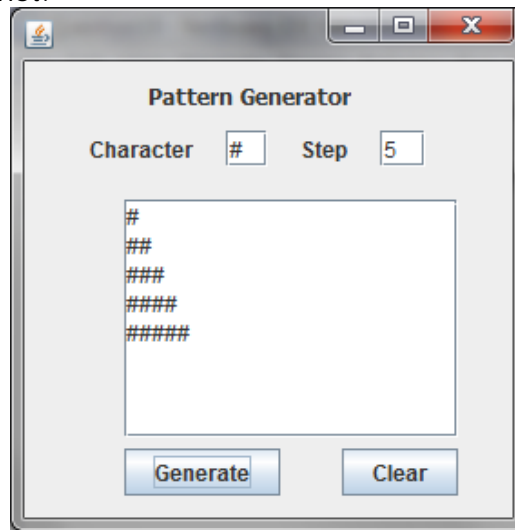
Experiment No. 8: (Problem 4.1 Page 165)

Objective: When a user selects a colour from list, the background of the appropriate control should be changed.

Experiment No. 9:

Objective: Understanding and use of Nested loops and Text Area control.

Task: Develop a Java application to print a Pattern for given character and steps, as per given screen shot.

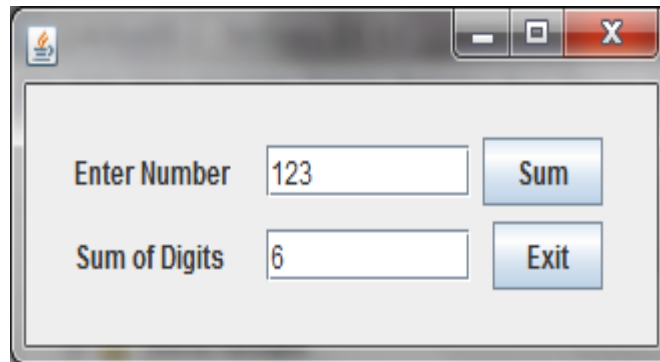


```
Coding: private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    String ch=TxtChar.getText();  
    int st=Integer.parseInt(TxtStep.getText());  
    for(int i=1;i<=st;i++)  
    {  
        for (int j = 1; j <= i; j++)  
        {  
            jTextArea1.append(ch);  
        }  
        jTextArea1.append("\n");  
    }  
}  
  
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    TxtChar.setText("");  
    TxtStep.setText("");  
    jTextArea1.setText("");  
}  
}
```


Experiment No. 10:

Objective: Understanding the use of loops and mathematical operations.

Task: Develop an application to compute the sum of digits for given number.

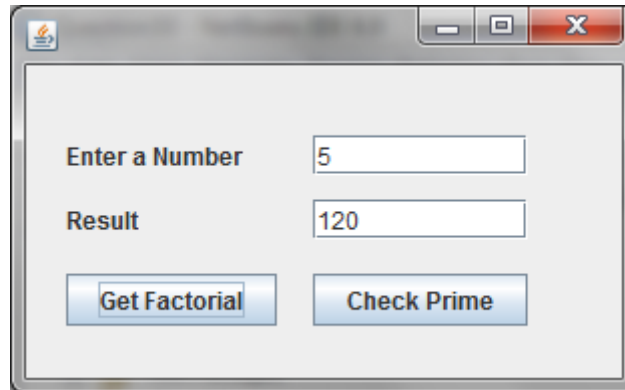


```
Coding: private void BtnSumActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    int num=Integer.parseInt(numTF.getText());  
    int sum=addDigits(num);  
    OutLbl.setText("Sum of digits: "+sum);  
}  
//User defined method to calculate sum of digits  
public int addDigits(int n)  
{ int s=0;  
  int dig;  
  while(n>0){  
    dig=n%10;  
    s=s+dig;  
    n=n/10;  
  }  
  return s;  
}  
private void BtnExitActionPerformed(java.awt.event.ActionEvent evt) {  
    System.exit(0);  
}
```

Experiment No. 11:

Objective: Understanding the use of User-defined methods in the application.

Task: Develop an application to compute the Factorial and Checking Prime for a given number, using custom methods. A method named factorial() and CheckPrime() along with suitable parameters are called when Get Factorial and Check Prime button is pressed respectively.



```
Coding: private void BtnFactActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    int n, fact;
    n=Integer.parseInt(TxtNumber.getText());
    fact=factorial(n);
    TxtResult.setText(""+fact);
}

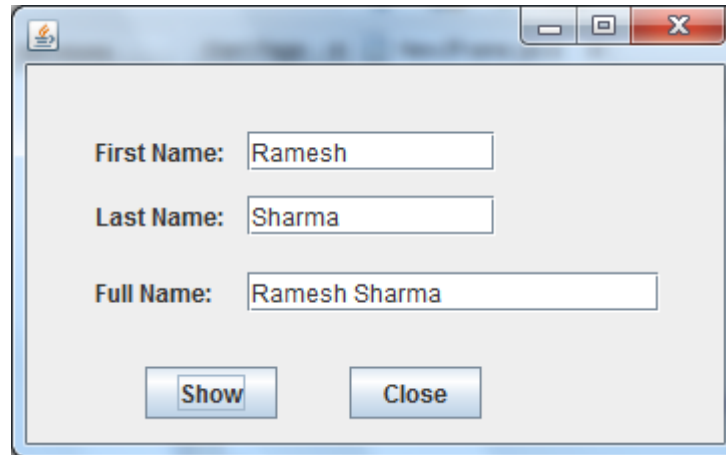
private void BtnPrimeActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    int n;
    n=Integer.parseInt(TxtNumber.getText());
    if (CheckPrime(n)==true)
        TxtResult.setText("Prime");
    else
        TxtResult.setText("Not Prime");
}

//User defined method to calculate factorial
public int factorial(int x)
{ int f=1;
  for(int i=1;i<=x;i++)
    f=f*i;
  return(f);
}

//User defined method to check prime or not.
public boolean CheckPrime(int x)
{ int i;
  for(i=2;i<x;i++)
  {
    if (x % 2 == 0)
      break;
  }
  if(i==x)
    return(true);
  else
    return(false);
}
```

Experiment No. 12:

Objective: To concatenate two strings.



```
Coding: private void BtnShowActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    TxtFullName.setText(TxtFName.getText() + " " + TxtLName.getText());  
}  
  
private void BtnCloseActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    System.exit(0);  
}
```

Experiment No. 13: (Problem 8, Page 223)

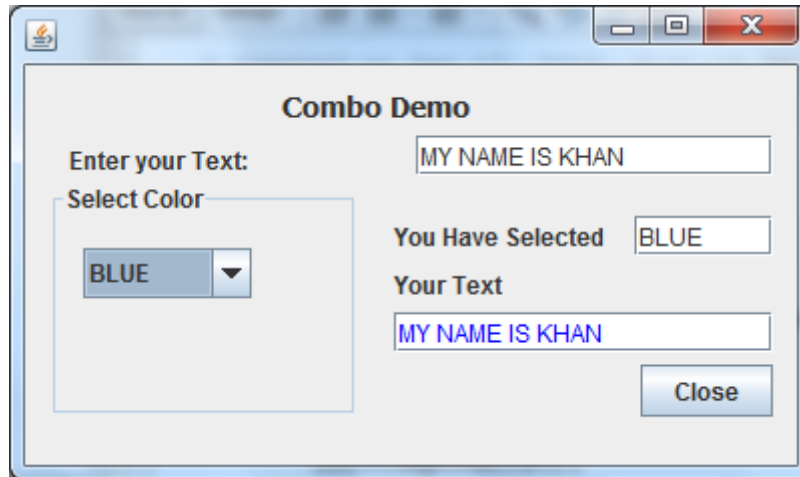
Objective: To replace a word in a given sentence.

Experiment No. 14: (Problem 2, Page 225)

Objective: To search a character in a given sentence.

Experiment No. 15:

Objective: Demonstration of use of Combo Box through code.



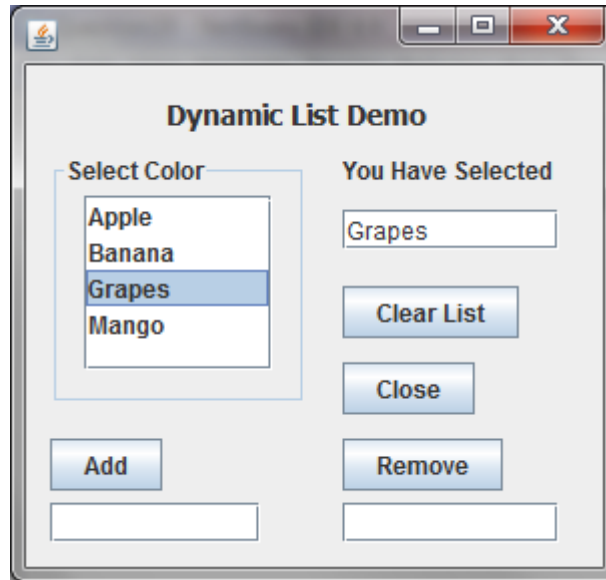
```
Coding: private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    System.exit(0);  
}
```

```
private void CboColorItemStateChanged(java.awt.event.ItemEvent evt) {  
    // TODO add your handling code here:  
    String text=TxtIn.getText();  
    String col=(String) CboColor.getSelectedItem();  
    TxtOut.setText(text);  
    TxtColor.setText(col);  
    switch (CboColor.getSelectedIndex())  
    { case 0: TxtOut.setForeground(Color.red);  
        break;  
      case 1: TxtOut.setForeground(Color.green);  
        break;  
      case 2: TxtOut.setForeground(Color.blue);  
        break;  
      case 3: TxtOut.setForeground(Color.magenta);  
        break;  
      case 4: TxtOut.setForeground(Color.yellow);  
        break;  
    }  
}
```

Experiment No. 16:

Objective: Demonstration of use of List Dynamically through code.

Task: Develop an application as per given screen shot to Add , Remove the given members of list and display the selected item in a text field using List control.



```
Coding: private void LstValueChanged(javax.swing.event.ListSelectionEvent evt) {
    // TODO add your handling code here:
    String text= (String) Lst.getSelectedValue();
    TxtOut.setText(text);
}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    System.exit(0);
}

private void BtnAddActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    DefaultListModel dlm=(DefaultListModel)Lst.getModel();
    dlm.addElement(TxtIn.getText());
    Lst.setModel(dlm);
    TxtIn.setText("");
}

private void BtnRemoveActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    DefaultListModel dlm=(DefaultListModel)Lst.getModel();
    dlm.removeElement(TxtRemove.getText());
    Lst.setModel(dlm);
}

private void BtnClearActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    DefaultListModel dlm=(DefaultListModel)Lst.getModel();
    dlm.removeAllElements();
    Lst.setModel(dlm);
}
```

Experiment No. 17: (Problem 1, Page 269)

Objective: Design a data connectivity application that fetches data from EMPL table and has an interface.

Experiment No. 18: (Problem 8.2, Page 256)

Objective: Updating/Deleting in the database.

Experiment No. 19:

Objective:

Design a JAVA application to navigate records in Student Table with following assumption.

Database Name : School (created in MySQL)

Table Name: Student (with few test records)

Table Structure : (Roll Integer , Name Char(3), Class Integer)

Password for MySQL: tiger

1. Design Java Application as per the screen shot.

Student Record

Roll No

Name

Class

Type and Name of Swing Control attached –

Type	Name	Purpose
JFrame	NewJFrame	Container Form to hold other GUI controls
jTextField	TxtRoll	To display Roll No.
	TxtName	To Display Name
	TxtClass	To Display Class
jButton	BtnFirst	To go on First Record
	BtnPrev	To go at Previous record
	BtnNext	To go at Next record
	BtnLast	To go at Last record
	BtnExit	To close the Application

Coding:

```
// Import Required Libraries
```

```
import java.sql.*;
```

```
import javax.swing.JOptionPane;
```

```
public class NewJFrame extends javax.swing.JFrame {  
    /* Global Variable declaration for connection, statement and ResultSet*/  
    Connection con=null;  
    Statement stmt=null;  
    ResultSet rs=null;  
    String DB="jdbc:mysql://localhost/school";
```

```
    /** Creates new form NewJFrame */
```

```
    public NewJFrame() {  
        initComponents();
```

```

/*Code to connect MySQL Database when application loads*/
try{
Class.forName("com.mysql.jdbc.Driver");
con=DriverManager.getConnection(DB,"root","tiger");
stmt=con.createStatement();
rs=stmt.executeQuery("select roll,name,class from student");

/* Locate Cursor on first Record when application loads */
rs.next();
TxtRoll.setText(""+rs.getInt("roll"));
TxtName.setText(""+rs.getString("name"));
TxtClass.setText(""+rs.getInt("class"));
}
catch (Exception e)
{ JOptionPane.showMessageDialog(null,"Error in Connection");    }
}

private void BtnExitActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:
// Coding to close connection and Application
try{
rs.close();
stmt.close();
con.close();
System.exit(0);
}
catch(Exception e)
{JOptionPane.showMessageDialog(null,"Unable to close connection");}
}

private void BtnFirstActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:
// Coding for Button First
try{
rs.first();
TxtRoll.setText(""+rs.getInt("roll"));
TxtName.setText(""+rs.getString("name"));
TxtClass.setText(""+rs.getInt("class"));
}
catch(Exception e)
{JOptionPane.showMessageDialog(null,"Error!!!");}
}

private void BtnPrevActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:
// Coding for Button Previous
try{
rs.previous();
if (rs.isBeforeFirst())
rs.last();
TxtRoll.setText(""+rs.getInt("roll"));
TxtName.setText(""+rs.getString("name"));
TxtClass.setText(""+rs.getInt("class"));
}
catch(Exception e)
{JOptionPane.showMessageDialog(null,"Error!!!");}
}

```



```

private void BtnNextActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    // Coding for Button Next
    try{
        rs.next() ;
        if (rs.isAfterLast())
            rs.first();
        TxtRoll.setText(""+rs.getInt("roll"));
        TxtName.setText(""+rs.getString("name"));
        TxtClass.setText(""+rs.getInt("class"));
    }
    catch(Exception e)
    {JOptionPane.showMessageDialog(null,"Error!!!");}
}

```

```

private void BtnLastActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    // Coding for Button Last
    try{
        rs.last() ;
        TxtRoll.setText(""+rs.getInt("roll"));
        TxtName.setText(""+rs.getString("name"));
        TxtClass.setText(""+rs.getInt("class"));
    }
    catch(Exception e)
    {JOptionPane.showMessageDialog(null,"Error!!!");}
}

```

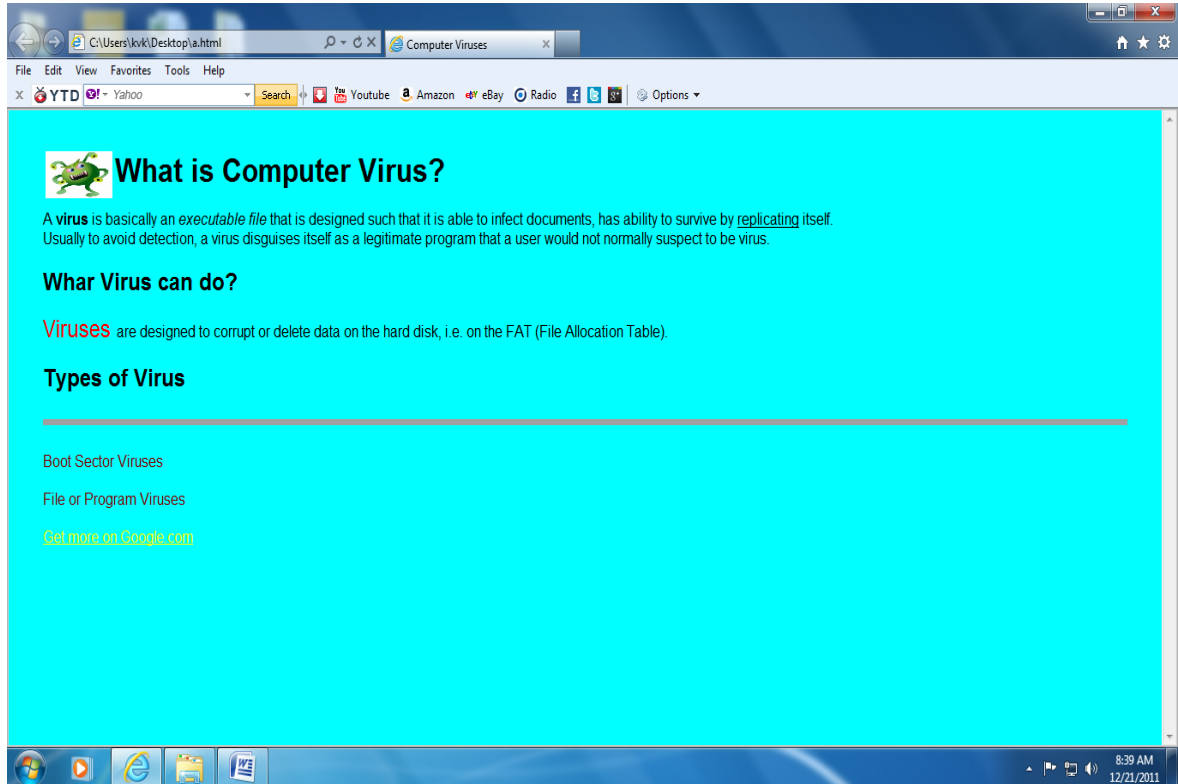
```

public static void main(String args[]) {
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new NewJFrame().setVisible(true);
        }
    });
}

```

Experiment No. 20:

Objective: Understanding the Web Page and use of different Tags and attributes.



```
<HTML>
<HEAD>
<TITLE> Computer Viruses</TITLE>
</HEAD>
<BODY BGCOLOR= #00ffff Topmargin=40 leftmargin=40>
<BASEFONT SIZE=3 FACE="Arial">
<IMG src="photo1.jpg" width="78" height="46" align="left">
<H1> What is Computer Virus?</H1>
<p align=left> A <b>virus</b> is basically an <i>executable file</i> that is designed
such that it is able to infect documents, has ability to survive by <u>replicating</u>
itself.<br> Usually to avoid detection, a virus disguises itself as a legitimate program that a
user would not normally suspect to be virus. </p>
<H2> Whar Virus can do? </H2>
<p> <font size=5 color= #ff0000 >Viruses </font>are designed to corrupt or delete data
on the hard disk, i.e. on the FAT (File Allocation Table).</p>
<H2> Types of Virus </H2>
<hr size=6 width=100% noshade>
<Font color= Maroon>
<p> Boot Sector Viruses </p>
<p> File or Program Viruses</p>
<a href="http://www.google.com/" ><font color="#ffff00" size=3> Get more on
Google.com</font></a>
</BODY>
</HTML>
```

Experiment No. 21:

Objective: Understanding the Web Form and use of different components to design an interactive form.

ADMISSION ENQUIRY FORM

Name :

Gender: Male Female

E-mail :

Stream:

Comment:


```
<html>
<head><title> My page </title> </Head>
<body>
<H1> <U>ADMISSSION ENQUIRY FORM </u></h1>
<Form method=Post action= "mailto:hashim_mbd@yahoo.co.in">
<b>Name </b> <Input type=Text name="st_name"><br>
<b>Gender </b>
<Input type=Radio name="gender" value="Male"> Male
<Input type=Radio name="gender" value="Female"> Female<Br>
</b>E-mail </B><Input type=Text Name ="email"> <br>
Stream <SELECT name="stream">
<Option value="Science"> Science </Option>
<Option value="Commerce"> Commerce </OPTION>
<Option value="Arts"> Arts </Option>
</SELECT> <Br>
Comment<Br>
<TextAREA name="comment" Rows=5 cols=50> </TEXTAREA><br>
<INPUT Type=Submit Value ="Send">
<INPUT Type=Reset Value ="Clear">
</Form>
</body>
</html>
```

Experiment No. 22

Objective: Understanding the use of MySQL queries.

- 1 Create and open Database named MYORG.
Ans: Create database MYORG;
Use MYORG;
- 2 Write a command to display the name of current month.
Ans: Select month(curdate());
- 3 Write commands to display the system date.
Ans: select sysdate;
- 4 Write a query to find out the result of 63.
Ans: select pow(6,3);
- 5 Write command to show the Tables in the MYORG Database.
Ans: Use MYORG;
Show tables;

DEPT

DeptID	DeptName	MgrID	Location
10	SALES	8566	Mumbai
20	PERSONEL	8698	Delhi
30	ACCOUNTS	8882	Delhi
40	RESEARCH	8839	Banglore

- 6 Add one column State of data type VARCHAR and size 30 to table DEPT
Ans: alter table DEPT
Add(state varchar(30));
- 7 Create a table name **EMP** with following structure

Column Name	EmpID	EmpName	Designation	DOJ	sal	comm	DeptID
Data Type	integer	Varchar (30)	Char(10)	Date	integer	integer	integer
Constraint	Primary Key	not null			Check > 1000		Foreign Key

Ans: create table emp
 (EmpID integer primary key,
 EmpName varchar(30) not null,
 Designation char(10),
 DOJ date,
 Sal integer check(sal>1000),
 Comm integer,
 DeptID integer,
 Foreign key(DeptID) references DEPT(DeptID));

EMP

EmpID	EmpName	Designation	DOJ	Sal	comm	DeptID
8369	SMITH	CLERK	18-12-1990	1050.00	200.00	10
8499	ANYA	SALESMAN	20-02-1991	1600.00	300.00	20
8566	MAHADEVAN	MANAGER	02-04-1991	2985.00	NULL	30
8654	MOMIN	SALESMAN	28-09-1991	1250.00	400.00	20
8698	BINA	MANAGER	05-01-1991	2850.00	250.00	30
8882	SHIVANSH	MANAGER	09-06-1991	2450.00	NULL	10
8888	SCOTT	ANALYST	09-12-1992	3000.00	150.00	10
8839	AMIR	PRESIDENT	18-11-1991	5000.00	NULL	20
8844	KULDEEP	SALESMAN	08-04-1992	1500.00	0.00	30

- 8 Insert the first record in table emp.
 Ans: insert into emp
 Values(8369,'SMITH','CLERK','18-12-1990',800,200,10);
- 9 Write a query to display EmpName and Sal of employees whose salary are greater than or equal to 2200
 Ans: select empname, sal
 from emp
 where sal >= 2200;
- 10 Write a query to display details of employees who are not getting commission.
 Ans: select *
 from emp
 where comm is NULL;
- 11 Write a query to display employee name and salary of those employees who don't have their salary in range of 2500 to 4000.
 Ans: select empname, sal
 From emp
 Where sal not between 2500 and 4000;

- 12 Write a query to display the name of employee whose name contains "A" as third alphabet in Ascending order of employee names.
Ans: select empname
From emp
Where empname like "__A%"
Order by empname;
- 13 Display the sum of salary and commission of employees as "Total Incentive" who are getting commission
Ans: select sal+comm As "Total Incentive"
From emp
where comm is not NULL;
- 14 Show the average salary for all departments with more than 5 working people.
Ans: select avg(sal)
From emp
Group by deptid
Having count(*)>5;
- 15 Display the distinct designation offered by the Organization.
Ans: select distinct designation
From emp;
- 16 List the count of Employees grouped by DeptID.
Ans: select count(*)
From emp
Group by DeptID;
- 17 Display the names of employees who joined on or after 01/05/1991.
Ans: Select empname
From emp
Where DOJ>='01/05/1991';
- 18 Display the employee records in order by DOJ.
Ans: select *
From emp
Order by DOJ;
- 19 Display the maximum salary of employees in each Department.
Ans: select max(sal)
From emp
Group by department;
20. Update all the records as add 'Mr.' with EmpName.
Ans: update emp
Set EmpName=concat('Mr',EmpName);

21. Display the name of Employees who is working in SALES department.
Ans: select empname
From emp, dept
Where deptName="SALES" and emp.DeptID=dept.DeptID;
22. Drop the emp table.
Ans: drop table emp;
23. Delete all the records who is working as "CLERK"
Ans: delete from emp
Where designation="CLERK";
24. Show the minimum, maximum and average salary of Managers.
Ans: select min(sal), max(sal), avg(sal)
From emp
Where designation="Manager";
25. Increase the salary of managers by 15%.
Ans: update emp
Set sal= sal + 0.15*sal
Where designation="Manager";
26. To display the name of those employees whose location is Delhi.
Ans: select empname
From emp, dept
Where location="Delhi" and emp.DeptID=dept.DeptID;
27. To display total salary of employees of sales department.
Ans: select sum(sal)
From emp, dept
Where deptName="SALES" and emp.DeptID=dept.DeptID;
28. To show details of employees who joined in the year 1991.
Ans: select *
From emp
Where year(DOJ)=1991;
29. Delete all the records who is working as "SALESMAN" and salary more than 1500.
Ans: delete from emp
Where designation=" SALESMAN" and sal>1500;
30. Set the commission as 100 who are not getting any commission.
Ans: update emp
Set comm= 100
Where comm Is NULL;