**Practical – 1**

**Aim: Create “Hello World” application to “Hello World” in the middle of the screen in the red color with white background.**

**Coding:**

activity\_main.xml

<?xml version="1.0" encoding="utf-8"?>

<android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context=".MainActivity">

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Hello World!"

android:textColor="@color/colorAccent"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintLeft\_toLeftOf="parent"

app:layout\_constraintRight\_toRightOf="parent"

app:layout\_constraintTop\_toTopOf="parent" />

</android.support.constraint.ConstraintLayout>

ActivityMain.class

package com.example.pro2;

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

public class MainActivity extends AppCompatActivity {

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

}

}

AndroidManifest.xml

<?xml version="1.0" encoding="utf-8"?>

<manifest xmlns:android="http://schemas.android.com/apk/res/android"

package="com.example.pro2">

<application

android:allowBackup="true"

android:icon="@mipmap/ic\_launcher"

android:label="@string/app\_name"

android:roundIcon="@mipmap/ic\_launcher\_round"

android:supportsRtl="true"

android:theme="@style/AppTheme">

<activity android:name=".MainActivity">

<intent-filter>

<action android:name="android.intent.action.MAIN" />

<category android:name="android.intent.category.LAUNCHER" />

</intent-filter>

</activity>

</application>

</manifest>

strings.xml

<resources>

<string name="app\_name">My Application</string>

</resources>

colors.xml

<?xml version="1.0" encoding="utf-8"?>

<resources>

<color name="colorPrimary">#008577</color>

<color name="colorPrimaryDark">#00574B</color>

<color name="colorAccent">#D81B60</color>

</resources>

**Output:**

**Practical – 2**

**Aim: Create sample application with login module. (Check username and password), validate it for login screen or alert the user with a Toast.**

**Coding:**

pro2.java

import java.applet.Applet ;

import java.awt.\*;

public class pro2 extends Applet {

public void init(){

this.setSize(300, 500);

} /\* Prepared By: Hardey Pandya \*/

public void paint(Graphics g){

g.setColor(Color.*red*);

for(int i=0 ; i<=500 ; i+=50)

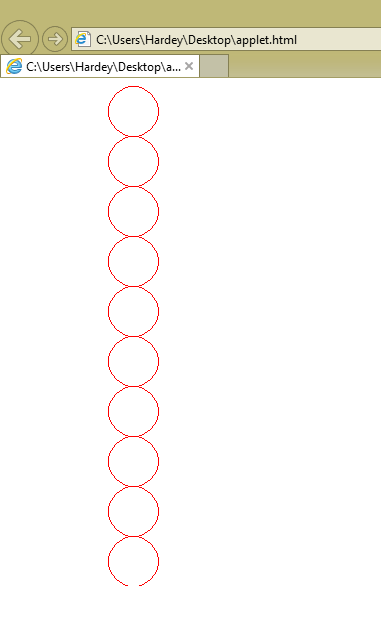
g.drawOval(100,i,50,50);

}

}

applet.html  
<applet code="pro2.class" height="500" width="500"/>

**Output:**



**Practical – 3**

**Aim:** **Create and validate a login application using username as Email ID else login button must remain disabled.**

**Coding:**

activity\_main.xml

<?xml version="1.0" encoding="utf-8"?>

<android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools"

android:id="@+id/tableLayout"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context=".MainActivity">

<TextView

android:id="@+id/textView4"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginStart="8dp"

android:layout\_marginEnd="8dp"

android:text="Password: "

app:layout\_constraintBottom\_toBottomOf="@+id/password\_text"

app:layout\_constraintEnd\_toStartOf="@+id/password\_text"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="@+id/password\_text"

app:layout\_constraintVertical\_bias="0.423" />

<TextView

android:id="@+id/textView3"

android:layout\_width="wrap\_content"

android:layout\_height="17dp"

android:layout\_marginTop="8dp"

android:layout\_marginEnd="8dp"

android:layout\_marginBottom="20dp"

android:text="Email ID: "

app:layout\_constraintBottom\_toTopOf="@+id/textView4"

app:layout\_constraintEnd\_toStartOf="@+id/email\_text"

app:layout\_constraintHorizontal\_bias="0.0"

app:layout\_constraintStart\_toStartOf="@+id/textView4"

app:layout\_constraintTop\_toTopOf="@+id/email\_text"

app:layout\_constraintVertical\_bias="0.092" />

<EditText

android:id="@+id/password\_text"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginEnd="21dp"

android:layout\_marginBottom="476dp"

android:ems="10"

android:inputType="textPassword"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toEndOf="@+id/textView4" />

<EditText

android:id="@+id/email\_text"

android:layout\_width="213dp"

android:layout\_height="0dp"

android:layout\_marginTop="44dp"

android:ems="10"

android:inputType="textEmailAddress"

android:textColorLink="@color/colorAccent"

app:layout\_constraintEnd\_toEndOf="@+id/password\_text"

app:layout\_constraintHorizontal\_bias="1.0"

app:layout\_constraintStart\_toStartOf="@+id/password\_text"

app:layout\_constraintTop\_toTopOf="parent" />

<Button

android:id="@+id/login\_button"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginTop="8dp"

android:layout\_marginBottom="408dp"

android:background="?attr/colorButtonNormal"

android:enabled="false"

android:text="Login"

android:textColor="?attr/colorControlNormal"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintStart\_toStartOf="@+id/textView4"

app:layout\_constraintTop\_toTopOf="parent"

app:layout\_constraintVertical\_bias="1.0" />

</android.support.constraint.ConstraintLayout>

MainActivity.java

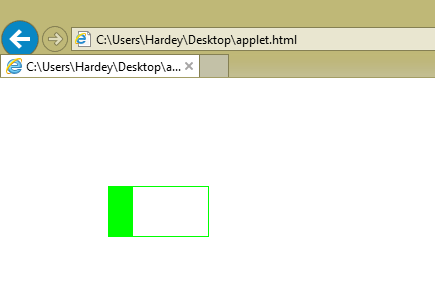
package com.example.pro2;  
import android.graphics.Color;  
import android.support.v7.app.AppCompatActivity;  
import android.os.Bundle;  
import android.text.Editable;  
import android.text.TextUtils;  
import android.text.TextWatcher;  
import android.util.Patterns;  
import android.widget.Button ;  
import android.widget.EditText;  
public class MainActivity extends AppCompatActivity implements TextWatcher {  
 EditText et1 ;  
 Button b1 ;  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*) ;  
 et1 = (EditText)findViewById(R.id.*email\_text*) ;  
 b1 = (Button)findViewById(R.id.*login\_button*) ;  
  
 et1.addTextChangedListener(this);  
 }

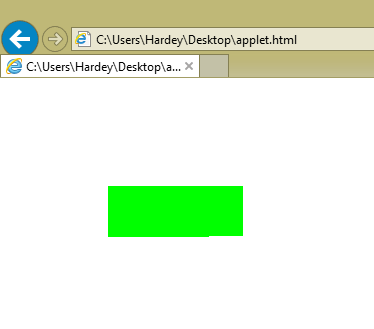
public static boolean isValidEmail(CharSequence target) {  
 return (!TextUtils.*isEmpty*(target) && Patterns.*EMAIL\_ADDRESS*.matcher(target).matches());  
 }  
  
 @Override  
 public void beforeTextChanged(CharSequence s, int start, int count, int after) {  
 if(MainActivity.*isValidEmail*(s)){  
 b1.setEnabled(true);  
 b1.setBackgroundColor(Color.*BLUE*);  
 }  
 else  
 b1.setEnabled(false);  
 }  
  
 @Override  
 public void onTextChanged(CharSequence s, int start, int before, int count) {  
 if(MainActivity.*isValidEmail*(s)){  
 b1.setEnabled(true);  
 b1.setBackgroundColor(Color.*BLUE*);  
 }  
 else  
 b1.setEnabled(false);  
 }  
  
 @Override  
 public void afterTextChanged(Editable s) {  
 if(MainActivity.*isValidEmail*(s)){  
 b1.setEnabled(true);  
 b1.setBackgroundColor(Color.*BLUE*);  
 }  
 else  
 b1.setEnabled(false);  
 }  
}

AndroidManifest.xml

*<?*xml version="1.0" encoding="utf-8"*?>*<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
 package="com.example.pro2">  
  
 <application  
 android:allowBackup="true"  
 android:icon="@mipmap/ic\_launcher"  
 android:label="@string/app\_name"  
 android:roundIcon="@mipmap/ic\_launcher\_round"  
 android:supportsRtl="true"  
 android:theme="@style/AppTheme">  
 <activity android:name=".MainActivity">  
 <intent-filter>  
 <action android:name="android.intent.action.MAIN" />  
  
 <category android:name="android.intent.category.LAUNCHER" />  
 </intent-filter>  
 </activity>  
 </application>  
  
</manifest>

**Output:**





**Practical – 4**

**Aim:** **Develop an applet that display the position of the mouse at the upper left corner of the applet when it is dragged or moved. Draw a 10x10 pixel rectangle filed with black at the current mouse position.**

**Coding:**

pro4.java

import java.applet.Applet;

import java.awt.Color;

import java.awt.Graphics;

import java.awt.event.MouseEvent;

import java.awt.event.MouseMotionListener;

public class pro4 extends Applet implements MouseMotionListener {

int x,y ;

public void start()

{

addMouseMotionListener(this);

}

public void paint(Graphics s)

{

s.drawString("Mouse Position:(" + x + "," + y + ")", 10, 10);

s.setColor(Color.black);

s.fillRect(x, y, 10, 10);

}

public void mouseDragged(MouseEvent arg0) {

x = arg0.getX( ) ;

y = arg0.getY( ) ;

paint(getGraphics());

repaint( ) ;

}

public void mouseMoved(MouseEvent arg0) {

x = arg0.getX( ) ;

y = arg0.getY( ) ;

paint(getGraphics());

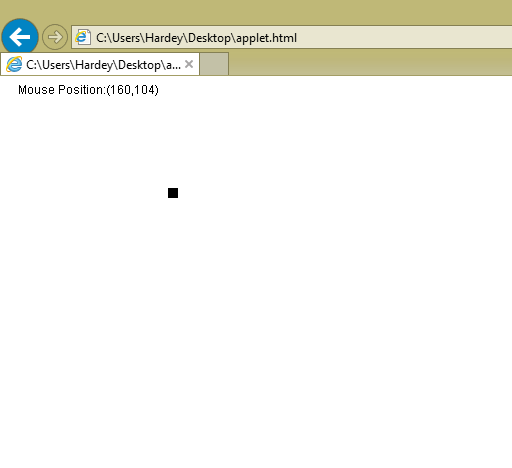
repaint( ) ;

}

}

applet.html  
<applet code="pro4.class" height="500" width="500"/>

**Output:**



**Practical – 5**

**Aim: Develop an applet that contains one button. Initialize the label on the button to “start”, when the user presses the button, which changes the label between these two values each time the button is pressed. Coding:**

pro5.java

import java.applet.Applet;

import java.awt.Button;

import java.awt.Color;

import java.awt.FlowLayout;

import java.awt.Graphics;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

public class pro5 extends Applet implements ActionListener {

static int i= 0 ;

Button b = new Button();

public void init()

{

setLayout(new FlowLayout());

add(b);

b.setVisible(true);

b.addActionListener(this);

b.setLabel("Start Hardey");

}

public void actionPerformed(ActionEvent arg0){

i++ ;

if(i%2 == 0)

b.setLabel("Start Hardey");

if(i%2 == 1)

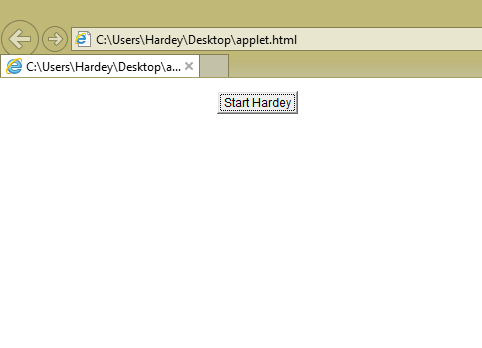
b.setLabel("End Pandya");

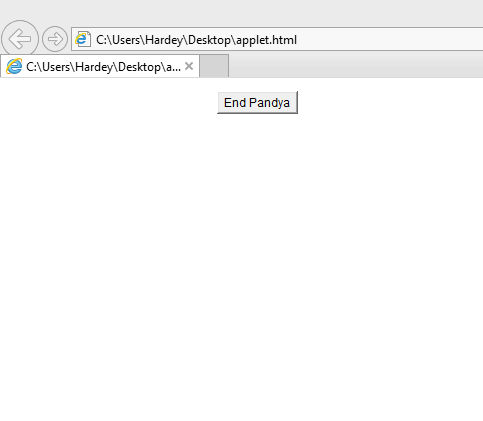
}

}

applet.html  
<applet code="pro5.class" height="500" width="500"/>

**Output:**





**Practical – 6**

**Aim:** **Develop an applet that uses the mouse listener, which overrides only two methods which are mousePressed and mouseReleased.**

**Coding:**

pro6.java

import java.applet.Applet;

import java.awt.Button;

import java.awt.Color;

import java.awt.FlowLayout;

import java.applet.Applet;

import java.awt.event.\*;

import java.awt.\*;

/\* Prepared By: Hardey Pandya \*/

public class pro6 extends Applet implements MouseListener{

static int var=-1;

public void init(){

this.setSize(500, 500);

this.addMouseListener( this);

}

public void mouseClicked(MouseEvent arg0) {

this.setBackground(Color.pink);

}

public void mouseEntered(MouseEvent arg0) {

this.setBackground(Color.black);

}

public void mouseExited(MouseEvent arg0) {

}

public void mousePressed(MouseEvent arg0) {

}

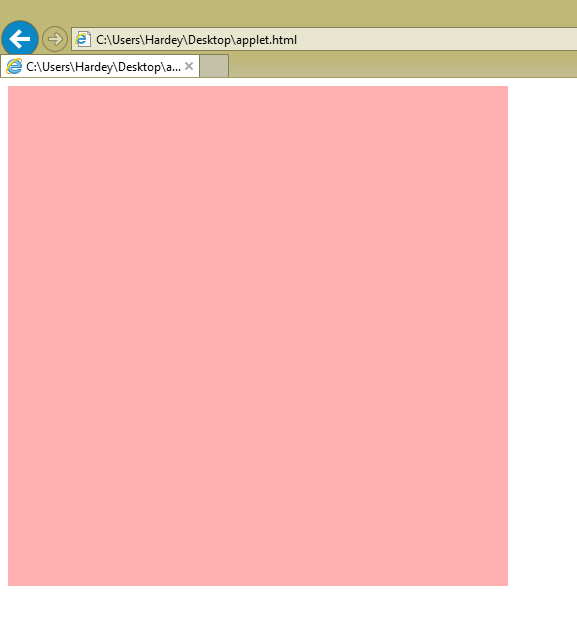
public void mouseReleased(MouseEvent arg0) {

}

}

applet.html  
<applet code="pro6.class" height="500" width="500"/>

**Output:**



**Practical – 7**

**Aim:** **Develop a program that has only one button in the frame, clicking on the button cycles through the colors: red->green->blue and so on. One color changes per click. (use getBackGround() method to get the current color)**

**Coding:**

import java.awt.\*;

import java.awt.event.\*;

public class pro7 extends Frame implements ActionListener{

static int var=-1;

pro7(){

this.setVisible(true);

Button b = new Button( ) ;

this.setSize(500, 500);

this.setLayout(new FlowLayout());

this.add(b) ;

b.addActionListener(this);

b.setLabel("Press Here Hardey Pandya");

}

public static void main(String args[]){

new pro7() ;

}

public void actionPerformed(ActionEvent arg0) {

var = (var+1) % 3 ;

if(var==0)

this.setBackground(Color.red);

else if(var==1)

this.setBackground(Color.green);

else if(var==2)

this.setBackground(Color.blue);

}

}

**Output:**



**Practical – 8**

**Aim:** **Develop an program that contains three check boxes and 30 x 30 pixel canvas.The three checkboxes should be labeled “Red”, “Green”,”Blue”. The selection of the check boxes determine the color of the canvas. For example, if the user selects both “Red” and “Blue”, the canvas should be purple.**

**Coding:**

import java.awt.\*;

I import java.awt.event.\*;

class MyCanvas extends Canvas implements ItemListener{

static int *r*=0;

static int *g*=0;

static int *b*=0;

public MyCanvas() {

setSize(300,300);

setBackground(Color.*black*);

}

public void itemStateChanged(ItemEvent arg0) {

if(arg0.getItem() == "red"){

if(arg0.getStateChange() == 1) *r* = 255 ;

else *r*=0 ;

}

if(arg0.getItem() == "green"){

if(arg0.getStateChange() == 1) *g* = 255 ;

else *g*=0 ;

}

if(arg0.getItem() == "blue"){

if(arg0.getStateChange() == 1) *b* = 255 ;

else *b*=0 ;

}

Color col = new Color(*r*,*g*,*b*) ;

this.setBackground(col);

}

}

public class pro8 extends Frame{

pro8(){

setTitle("HardeyPandya");

this.setVisible(true);

Checkbox c1 = new Checkbox("red") ;

Checkbox c2 = new Checkbox("green") ;

Checkbox c3 = new Checkbox("blue") ;

this.setSize(500, 350);

this.setLayout(new FlowLayout());

this.add(c1) ;

this.add(c2) ;

this.add(c3) ;

MyCanvas mycan = new MyCanvas() ;

add(mycan) ;

c1.addItemListener(mycan);

c2.addItemListener(mycan);

c3.addItemListener(mycan);

}

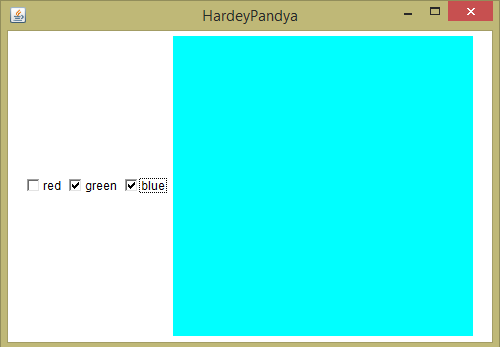
public static void main(String args[]){

new pro8() ;

}

}

**Output:**



**Practical – 9**

**Aim: Create an application that displays a frame with a menu bar. When a user selects any menu or menu item, display that selection on a text area in the center of the frame.**

**Coding:**

import java.awt.\*;

import java.awt.event.\*;

public class pro9 implements ActionListener

{

Frame f;

Label l;

pro9()

{

f = new Frame("Hardey Pandya") ;

l= new Label("Select Menu.");

MenuBar mb = new MenuBar();

Menu mn = new Menu("File");

MenuItem mi1 = new MenuItem("New");

mi1.addActionListener(this);

MenuItem mi2 = new MenuItem("Open");

mi2.addActionListener(this);

MenuItem mi3 = new MenuItem("Save");

mi3.addActionListener(this);

MenuItem mi4 = new MenuItem("Exit");

mi4.addActionListener(this);

mn.add(mi1);

mn.add(mi2);

mn.add(mi3);

mn.add(mi4);

mb.add(mn);

f.setMenuBar(mb);

f.setLayout(new FlowLayout());

f.add(l);

f.setSize(300,300);

f.setVisible(true);

}

public static void main(String[] args){

pro9 f = new pro9();

}

public void actionPerformed(ActionEvent ae){

if(ae.getSource() instanceof MenuItem){

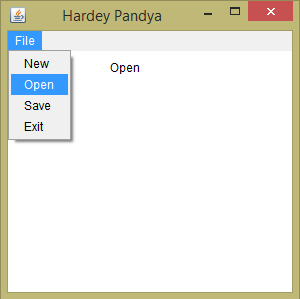
l.setText(ae.getActionCommand());

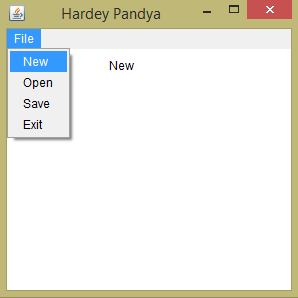
}

}

}

**Output:**





**Practical – 10**

**Aim: Develop a program that draws two sets of ever-decreasing rectangles one in outline form and one filled alternately in black and white.**

**Coding:**

import java.awt.\*;

public class pro10 extends Frame {

pro10(){

setTitle("HardeyPandya") ;

setSize(900, 600) ;

setVisible(true) ;

}

public void paint(Graphics g){

int x1 = 40, y1 = 215, w1 = 320, h1 = 170 ;

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

g.drawRect(x1+=10, y1+=10, w1-=20, h1-=20) ;

int x2 = 490, y2 = 215, w2 = 320, h2 = 170 ;

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

if (i % 2 == 0) {

g.setColor(Color.BLACK);

g.fillRect(x2 += 10, y2 += 10, w2 -= 20, h2 -= 20);

}

else {

g.setColor(Color.WHITE);

g.fillRect(x2 += 10, y2 += 10, w2 -= 20, h2 -= 20);

}

}

}

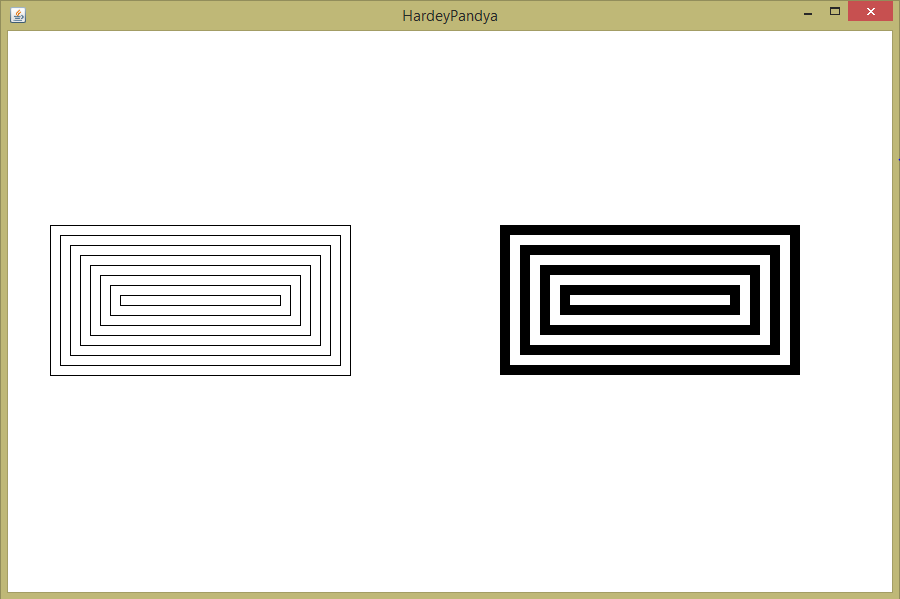
public static void main(String ma[]){

new pro10() ;

}

}

**Output:**



**Practical – 11**

**Aim: Develop a database application that uses any JDBC driver**

**Coding:**

import java.sql.\*;

public class pro11 {

Statement st ;

Connection cn ;

public pro11() {

try {

Class.forName("com.mysql.cj.jdbc.Driver") ;

Connection cn =

DriverManager.getConnection("jdbc:mysql://localhost/","root","");

st = cn.createStatement() ;

st.executeUpdate("use dat;") ;

st.executeUpdate("create table hardeypandya ( nam varchar(50),rolno

int(10) ); " ) ;

}

catch (Exception e) {

e.printStackTrace();

}

}

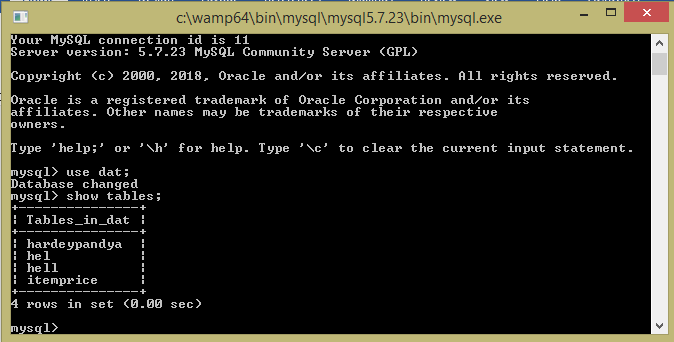
public static void main(String ar[]) {

new pro11( ) ;

}

}

**Output:**



**Practical – 12**

**Aim: Develop a Graphical User Interface that performs the following SQL operations: a) Insert b) Delete c) Update.**

**Coding:**

import java.awt.EventQueue;

import javax.swing.JFrame;

import javax.swing.JOptionPane;

import javax.swing.JPanel;

import javax.swing.border.EmptyBorder;

import javax.swing.JLabel;

import java.awt.Color;

import javax.swing.JTextField;

import javax.swing.JButton;

import java.awt.event.ActionListener;

import java.awt.event.ActionEvent;

import java.sql.\*;

public class pro12 extends JFrame {

private JPanel contentPane;

private JTextField textField;

private JTextField textField\_1;

private JTextField textField\_2;

private JTextField textField\_3;

private JTextField textField\_4;

Connection con ;

Statement st ;

ResultSet rs ;

JLabel label = new JLabel("");

String dat ;

public static void main(String[] args) {

EventQueue.*invokeLater*(new Runnable() {

public void run() {

try {

pro12 frame = new pro12();

frame.setVisible(true);

} catch (Exception e) {

e.printStackTrace();

}

}

});

}

public pro12() {

setVisible(true);

try {

Class.*forName*("com.mysql.cj.jdbc.Driver") ;

con=DriverManager.*getConnection*("jdbc:mysql://localhost/","root","");

st = con.createStatement();

st.executeUpdate("use dat") ;

}

catch (Exception e) {

JOptionPane.*showMessageDialog*(null,"Please connect to database.");

}

setTitle("Operations in Database 'dat'");

setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);

setBounds(100, 100, 701, 357);

contentPane = new JPanel();

contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

contentPane.setLayout(null);

showdat();

JLabel lblInsert = new JLabel("Insert:");

lblInsert.setForeground(Color.*RED*);

lblInsert.setBounds(10, 11, 46, 14);

contentPane.add(lblInsert);

JLabel lblDelete = new JLabel("Delete:");

lblDelete.setForeground(Color.*RED*);

lblDelete.setBounds(10, 121, 46, 14);

contentPane.add(lblDelete);

JLabel lblUpdate = new JLabel("Update:");

lblUpdate.setForeground(Color.*RED*);

lblUpdate.setBounds(10, 201, 46, 14);

contentPane.add(lblUpdate);

JLabel lblColumn = new JLabel("nam(string):");

lblColumn.setForeground(Color.*BLACK*);

lblColumn.setBounds(20, 36, 94, 14);

contentPane.add(lblColumn);

JLabel lblColumnint = new JLabel("rol no(int):");

lblColumnint.setForeground(Color.*BLACK*);

lblColumnint.setBounds(20, 62, 94, 14);

contentPane.add(lblColumnint);

JLabel lblUpdateColumnName = new JLabel("Enter old nam:");

lblUpdateColumnName.setForeground(Color.*BLACK*);

lblUpdateColumnName.setBounds(20, 226, 116, 14);

contentPane.add(lblUpdateColumnName);

JLabel lblUpdateColumnName\_1 = new JLabel("Update nam:");

lblUpdateColumnName\_1.setForeground(Color.*BLACK*);

lblUpdateColumnName\_1.setBounds(20, 251, 116, 14);

contentPane.add(lblUpdateColumnName\_1);

JLabel lblRowNumber = new JLabel("Enter nam to delete:");

lblRowNumber.setForeground(Color.*BLACK*);

lblRowNumber.setBounds(20, 146, 113, 14);

contentPane.add(lblRowNumber);

textField = new JTextField();

textField.setBounds(143, 33, 220, 20);

contentPane.add(textField);

textField.setColumns(10);

textField\_1 = new JTextField();

textField\_1.setColumns(10);

textField\_1.setBounds(143, 59, 220, 20);

contentPane.add(textField\_1);

textField\_2 = new JTextField();

textField\_2.setColumns(10);

textField\_2.setBounds(143, 143, 220, 20);

contentPane.add(textField\_2);

textField\_3 = new JTextField();

textField\_3.setColumns(10);

textField\_3.setBounds(143, 223, 220, 20);

contentPane.add(textField\_3);

textField\_4 = new JTextField();

textField\_4.setColumns(10);

textField\_4.setBounds(143, 248, 220, 20);

contentPane.add(textField\_4);

JButton btnNewButton = new JButton("Insert");

btnNewButton.setBounds(10, 87, 89, 23);

contentPane.add(btnNewButton);

btnNewButton.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent arg0) {

try {

st.executeUpdate("insert into hell values ( '" + textField.getText()

+ "',"+ Integer.*parseInt*(textField\_1.getText()) + ");" ) ;

showdat();

}

catch (Exception e) {

JOptionPane.*showMessageDialog*(null,"Please Enter appropriate

data in both fields");

e.printStackTrace();

}

}

});

JButton btnDelete = new JButton("Delete");

btnDelete.setBounds(10, 167, 89, 23);

contentPane.add(btnDelete);

btnDelete.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent arg0) {

try {

st.executeUpdate("delete from hell where ( nam='" +

textField\_2.getText() + "');" ) ;

showdat();

} catch (SQLException e) {

JOptionPane.*showMessageDialog*(null,"Please Enter nam that

exist.");

}

}

});

JButton btnUpdate = new JButton("Update");

btnUpdate.setBounds(10, 276, 89, 23);

contentPane.add(btnUpdate);

label.setBounds(436, 11, 239, 281);

contentPane.add(label);

btnUpdate.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent arg0) {

try {

st.executeUpdate("UPDATE hell SET nam ='" +

textField\_4.getText() + "' WHERE nam ='" + textField\_3.getText() + "';" ) ;

showdat();

} catch (SQLException e) {

JOptionPane.*showMessageDialog*(null,"Please Enter nam that

exist.");

}

}

});

}

public void showdat(){

try {

rs = st.executeQuery("select \* from hell") ;

dat = "<html>" ;

rs.absolute(0) ;

while(rs.next())

dat = dat + rs.getString(1) + " " + rs.getInt(2) + "<br> " ;

dat = dat + "</html>" ;

label.setText(dat);

} catch (SQLException e) {

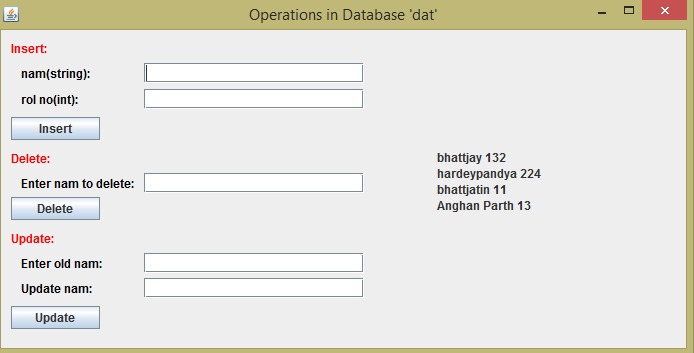
e.printStackTrace();

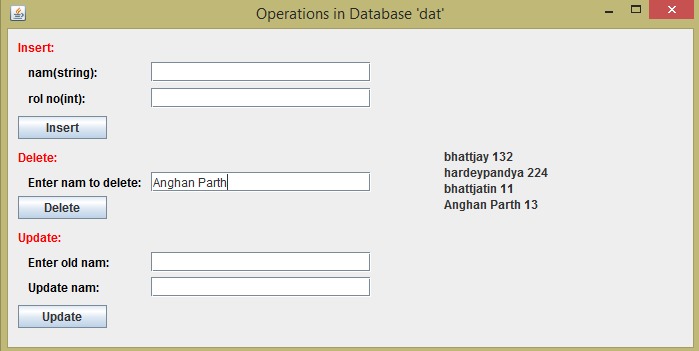
}

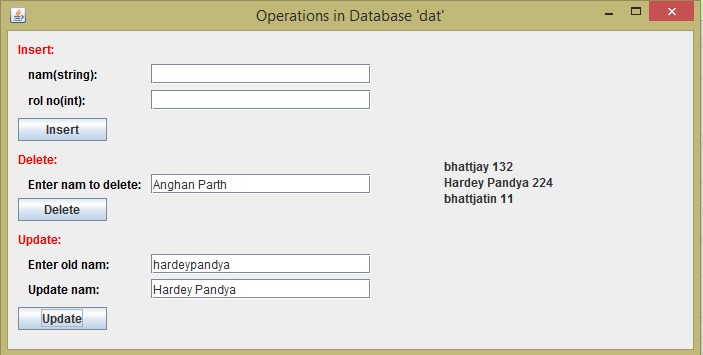
}

}

**Output:**







**Practical – 13**

**Aim: Develop a program to present a set of choice for user to select a product and display the price of product.**

**Coding:**

import java.awt.BorderLayout;

import java.awt.EventQueue;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JOptionPane;

import javax.swing.JPanel;

import javax.swing.border.EmptyBorder;

import com.mysql.cj.x.protobuf.MysqlxCrud.Insert;

import javax.swing.JComboBox;

import javax.swing.JButton;

import java.awt.event.ActionListener;

import java.awt.event.ActionEvent;

import javax.swing.JTextField;

public class frm13 extends JFrame implements ActionListener {

private JPanel contentPane;

private String p ;

Connection con ;

Statement st ;

ResultSet rs ;

String dat ;

String Item[]={"Pani Puri","Pav Bhaji","Bhel","Dabeli","Tea","Coffee","Hot Milk","Hot

Chocolate"};

private JTextField textField = new JTextField();

private JTextField textField\_1;

JComboBox comboBox\_1 ;

private JLabel lblHardeyPandya;

public static void main(String[] args) {

EventQueue.invokeLater(new Runnable() {

public void run() {

try {

frm13 frame = new frm13();

frame.setVisible(true);

} catch (Exception e) {

e.printStackTrace();

}

}

});

}

public frm13() {

try {

Class.forName("com.mysql.cj.jdbc.Driver") ;

con=DriverManager.getConnection("jdbc:mysql://localhost/","root","");

st = con.createStatement();

st.executeUpdate("use dat;") ;

} catch (Exception e) {

JOptionPane.showMessageDialog(null,"Please connect to database.");

}

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setBounds(100, 100, 281, 137);

contentPane = new JPanel();

contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));

contentPane.setLayout(new BorderLayout(0, 0));

setContentPane(contentPane);

JButton btnClickHereTo = new JButton("Click Here to see the Price");

contentPane.add(btnClickHereTo, BorderLayout.WEST);

comboBox\_1 = new JComboBox(Item);

contentPane.add(comboBox\_1, BorderLayout.NORTH);

textField\_1 = new JTextField();

contentPane.add(textField\_1, BorderLayout.SOUTH);

textField\_1.setColumns(10);

lblHardeyPandya = new JLabel("Hardey");

contentPane.add(lblHardeyPandya, BorderLayout.EAST);

btnClickHereTo.addActionListener(this) ;

}

public void actionPerformed(ActionEvent arg0) {

try {

textField\_1.setText("");

rs = st.executeQuery("Select price from ITEMPRICE where item= '" +

comboBox\_1.getItemAt(comboBox\_1.getSelectedIndex()) + "' ;") ;

rs.next() ;

int pi = rs.getInt(1) ;

textField\_1.setText(String.valueOf(pi));

}

catch (Exception e) {

e.printStackTrace();

}

}

}

**Output:**

