PART-C

# Write a program that creates a user interface to perform basic integer operations. The user enters two numbers in the textfields, Num1 and Num2. The result of operations must be displayed in the Result textfield when the “=” button is clicked. If Num1 or Num2 is not an integer, the program should throw NumberFormatException. If Num2 is Zero, the program should throw an ArithmeticException when division operation is applied. Display the exception in a message dialog box. [Swing]

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

class C\_1\_Arithmetic extends JFrame implements ActionListener

{

JTextField v1,op,v2,res;

JButton b;

JLabel l1,l2,l3,l4;

C\_1\_Arithmetic()

{

Container c=getContentPane();

c.setLayout(new FlowLayout());

l1=new JLabel("Number 1");

l2=new JLabel("Operation");

l3=new JLabel("Number 2");

l4=new JLabel("Result");

v1=new JTextField("",5);

op=new JTextField("",1);

v2=new JTextField("",5);

res=new JTextField("",5);

b=new JButton("=");

c.add(l1);

c.add(v1);

c.add(l2);

c.add(op);

c.add(l3);

c.add(v2);

c.add(b);

c.add(l4);

c.add(res);

b.addActionListener(this);

}

public void actionPerformed(ActionEvent ae)

{

if(ae.getActionCommand()=="=")

{

try

{

char operator=op.getText().charAt(0);

int r=0,num1=0,num2=0;

num1=Integer.parseInt(v1.getText());

num2=Integer.parseInt(v2.getText());

switch(operator)

{

case '+': r=num1+num2;

break;

case '-': r=num1-num2;

break;

case '\*': r=num1\*num2;

break;

case '/': r=num1/num2;

break;

}

res.setText(""+r);

}

catch(ArithmeticException e1)

{

JOptionPane.showMessageDialog(null,"ArithmeticException");

res.setText("");

}

catch(NumberFormatException e1)

{

JOptionPane.showMessageDialog(null,"NumberFormatException");

res.setText("");

}

}

}

public static void main(String args[])

{

C\_1\_Arithmetic ob = new C\_1\_Arithmetic();

ob.setSize(800,600);

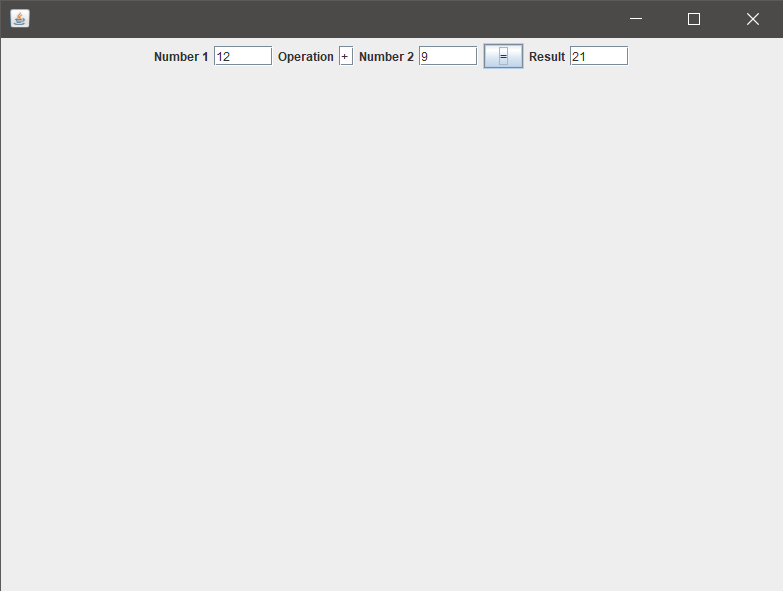
ob.setVisible(true);

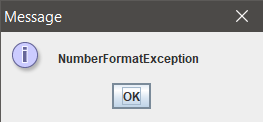
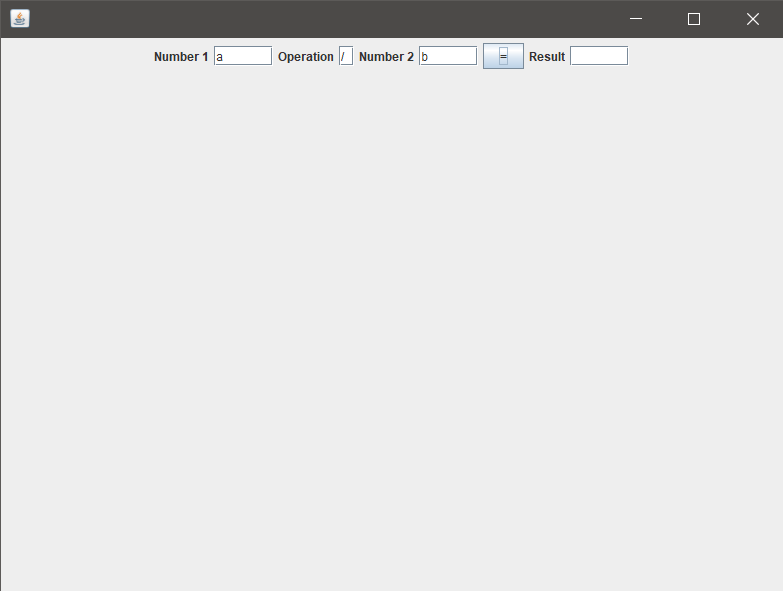
ob.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

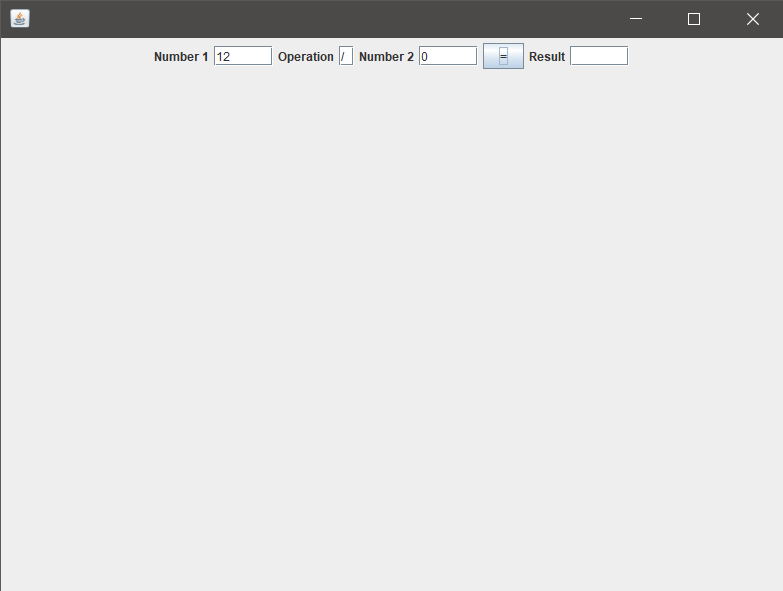
}

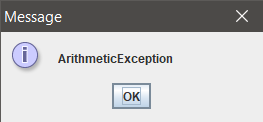
}

Output:









# Write a program to simulate a Traffic Light. The program lets the user select one of three lights: red, yellow or Green with radio buttons. On selecting radio button, an appropriate message with “Stop” or “Ready” or “Go” should appear above the button in selected color. Initially, there is no message shown. [Swing]

import javax.swing.\*;

import javax.swing.event.\*;

import java.awt.\*;

import java.awt.event.\*;

class Light extends JFrame implements ItemListener

{

public JLabel l1,l2;

public JRadioButton r1,r2,r3;

public ButtonGroup bg;

public JPanel p,p1;

public Light()

{

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setLayout(new GridLayout(2, 1));

setSize(800,400);

setTitle("TRAFFIC");

p= new JPanel(new FlowLayout());

p1= new JPanel(new FlowLayout());

l1= new JLabel();

Font f=new Font("Verdana",Font.BOLD,60);

l1.setFont(f);

add(l1);

p.add(l1);

add(p);

l2=new JLabel("Select Lights");

p1.add(l2);

JRadioButton r1 = new JRadioButton("Red Light");

r1.setBackground(Color.red);

p1.add(r1);

r1.addItemListener(this);

JRadioButton r2 = new JRadioButton("Yellow Light");

r2.setBackground(Color.YELLOW);

p1.add(r2);

r2.addItemListener(this);

JRadioButton r3 = new JRadioButton("Green Light");

r3.setBackground(Color.GREEN);

p1.add(r3);

r3.addItemListener(this);

add(p1);

bg=new ButtonGroup();

bg.add(r1);

bg.add(r2);

bg.add(r3);

setVisible(true);

}

public void itemStateChanged(ItemEvent i)

{

JRadioButton jb= (JRadioButton)i.getSource();

switch(jb.getText())

{

case "Red Light":

l1.setText("STOP");

l1.setForeground(Color.RED);

break;

case "Yellow Light":

l1.setText("Ready");

l1.setForeground(Color.YELLOW);

break;

case "Green Light":

l1.setText("GO");

l1.setForeground(Color.GREEN);

break;

}

}

}

public class C\_2\_Traffic

{

public static void main(String args[])

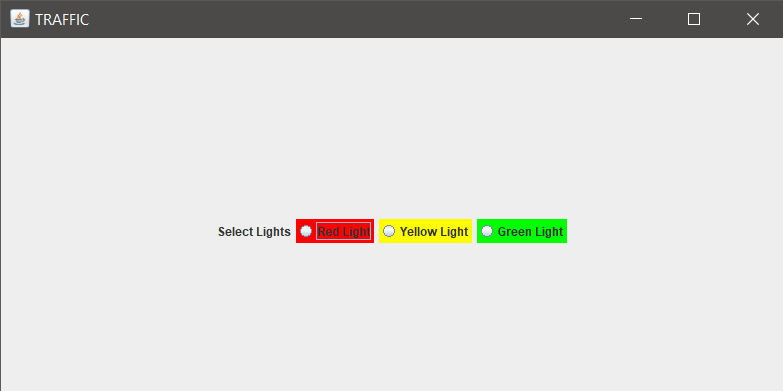
{

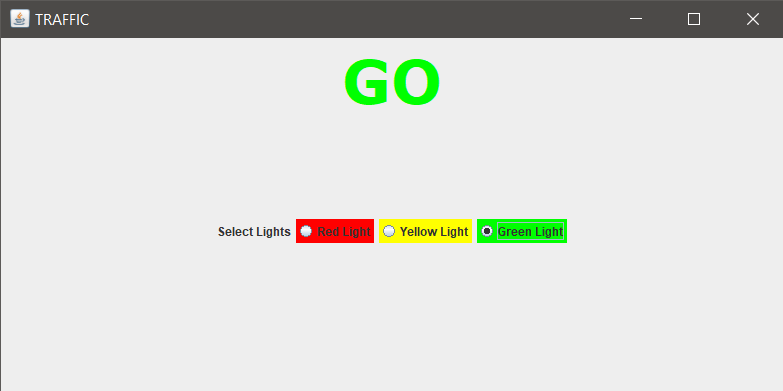
Light a= new Light();

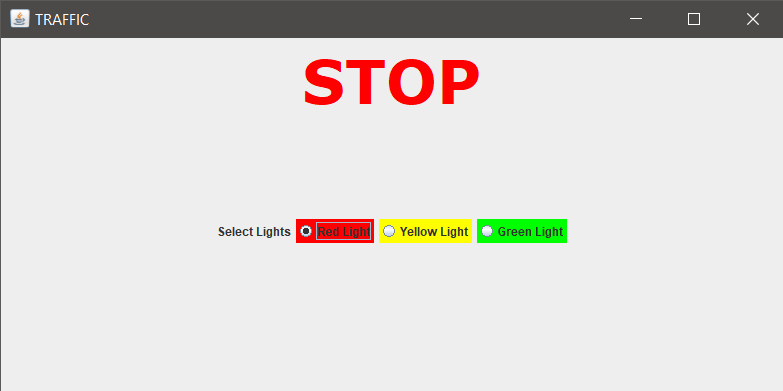
}

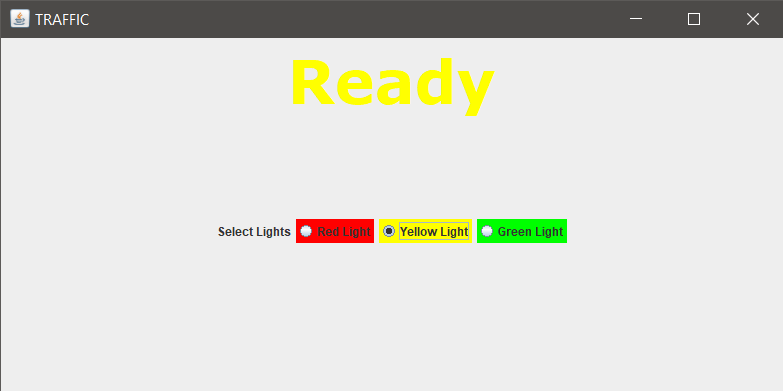
}

Output:









# 3. Write a menu driven JDBC program to perform basic operations with Student Table. Operations to performed are insert student details, delete a specific student details and search for a student’s details. [JDBC]

# 4. Write a program to design a registration form for creating a new eMail account. [Swing &JDBC]

# 5. Write a program to retrieve data from telephone table (fname, lastname, telNo) and display them in a JTable component. [Swing & JDBC]