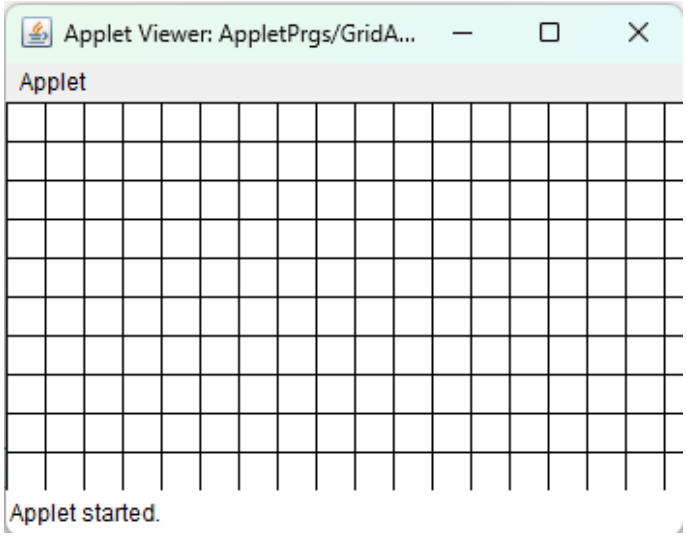
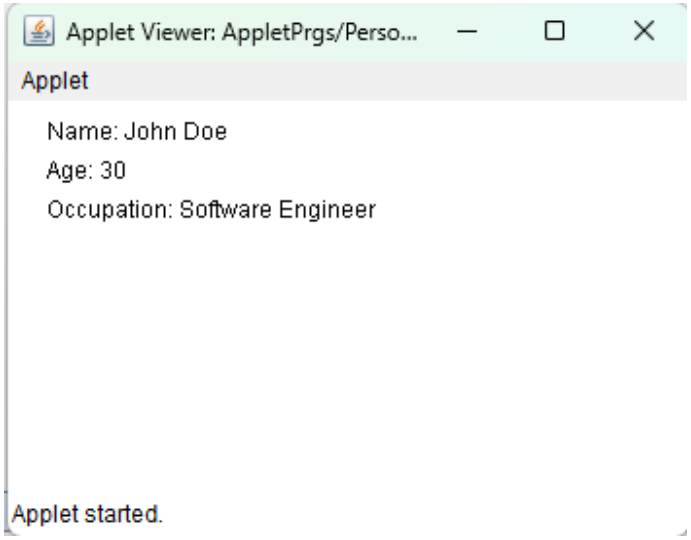
15. Write a program to create an applet and draw grid lines

```
import java.applet.Applet;
import java.awt.*;

public class GridLinesApplet extends Applet {
    public void paint(Graphics g) {
        int width = getSize().width;
        int height = getSize().height;

        // Set color for grid lines
        g.setColor(Color.BLACK);

        // Draw vertical lines
        for (int i = 0; i < width; i += 20) {
            g.drawLine(i, 0, i, height);
        }

        // Draw horizontal lines
        for (int i = 0; i < height; i += 20) {
            g.drawLine(0, i, width, i);
        }
    }
}
```



16. Create a simple applet which reveals the personal information of yours.

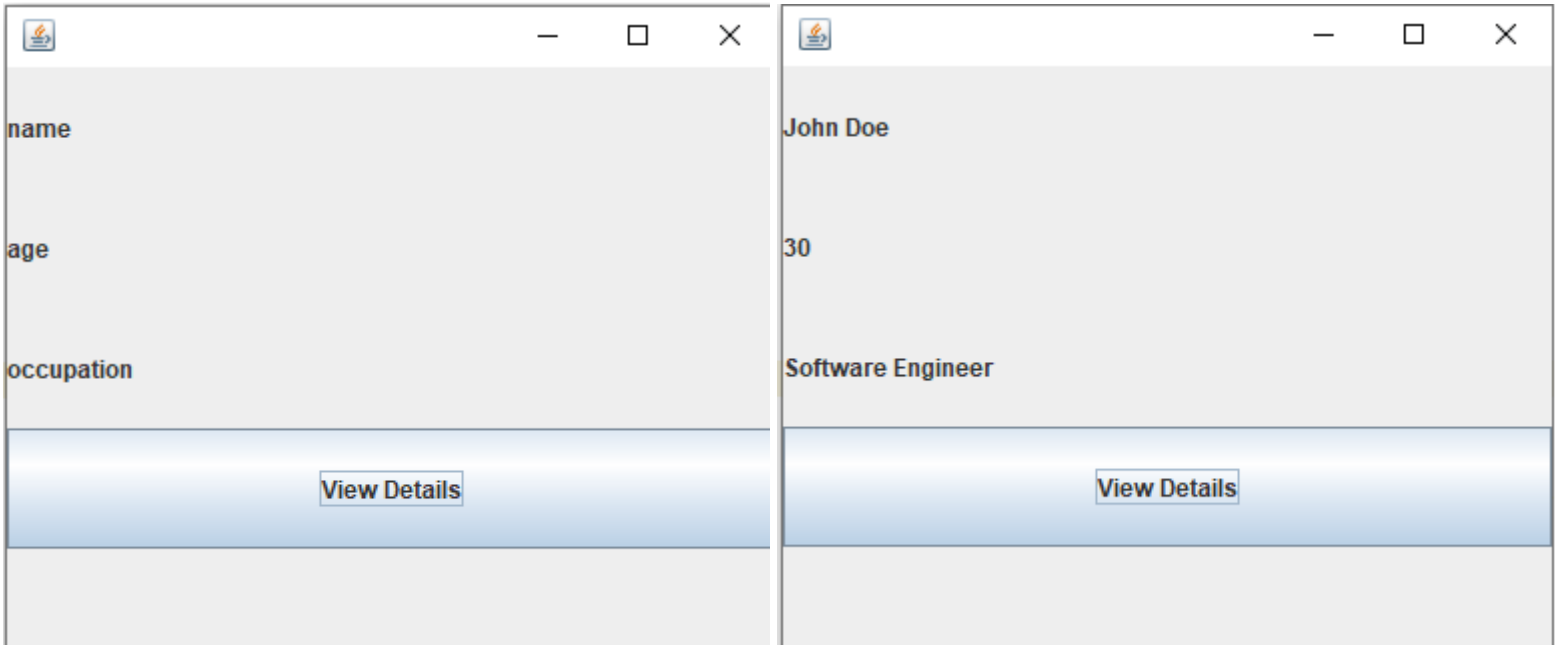
```
import java.applet.Applet;
import java.awt.*;

public class PersonalInfoApplet extends Applet {
    private String name;
    private int age;
    private String occupation;

    @Override
    public void init() {
        name = "John Doe";
        age = 30;
        occupation = "Software Engineer";
    }

    @Override
    public void paint(Graphics g) {
        g.drawString("Name: " + name, 20, 20);
        g.drawString("Age: " + age, 20, 40);
        g.drawString("Occupation: " + occupation, 20, 60);
    }
}
```



17. Create a frame which displays your personal details with respect to a button click

```
import java.awt.*;
import javax.swing.*;
import java.awt.event.*;

public class PersonalDetails extends JFrame implements ActionListener {

    public PersonalDetails() {

        name = new JLabel();
        age = new JLabel();
        occupation = new JLabel();
        viewDetails = new JButton();

        name.setText("name");
        age.setText("age");
        occupation.setText("occupation");
        viewDetails.setText("View Details");

        setLayout(new GridLayout(6, 1));

        add(name);
        add(age);
        add(occupation);
        add(viewDetails);

        setSize(400, 400);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setVisible(true);

        viewDetails.addActionListener(this);
    }

    public void actionPerformed(ActionEvent evt) {
        name.setText("John Doe");
        age.setText("30");
        occupation.setText("Software Engineer");

    }
```

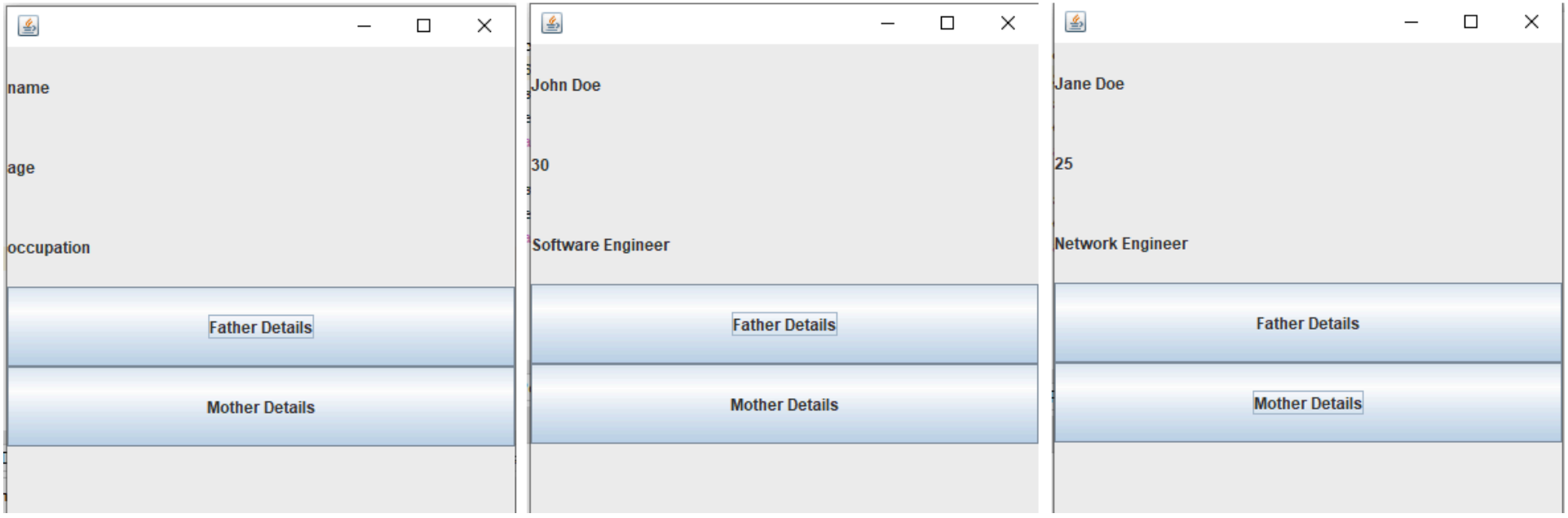
```
    public static void main(String args[]) {
        new PersonalDetails();
    }

    JLabel name, age, occupation;
    JButton viewDetails;
}
```



18. Write a program which creates a frame with two buttons: father and mother. When we click the father button the name of the father, his age and designation must appear. When we click mother, similar details of mother also appear.

```
import java.awt.*;
import javax.swing.*;
import java.awt.event.*;

public class PersonalDetails extends JFrame implements ActionListener {

    public PersonalDetails() {

        name = new JLabel();
        age = new JLabel();
        occupation = new JLabel();
        fatherDetails = new JButton();
        motherDetails = new JButton();

        name.setText("name");
        age.setText("age");
        occupation.setText("occupation");
        fatherDetails.setText("Father Details");
        motherDetails.setText("Mother Details");

        setLayout(new GridLayout(6, 1));

        add(name);
        add(age);
        add(occupation);
        add(fatherDetails);
        add(motherDetails);

        setSize(400, 400);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setVisible(true);

        fatherDetails.addActionListener(this);
        motherDetails.addActionListener(this);
    }

    public static void main(String args[]) {
        new PersonalDetails();
```

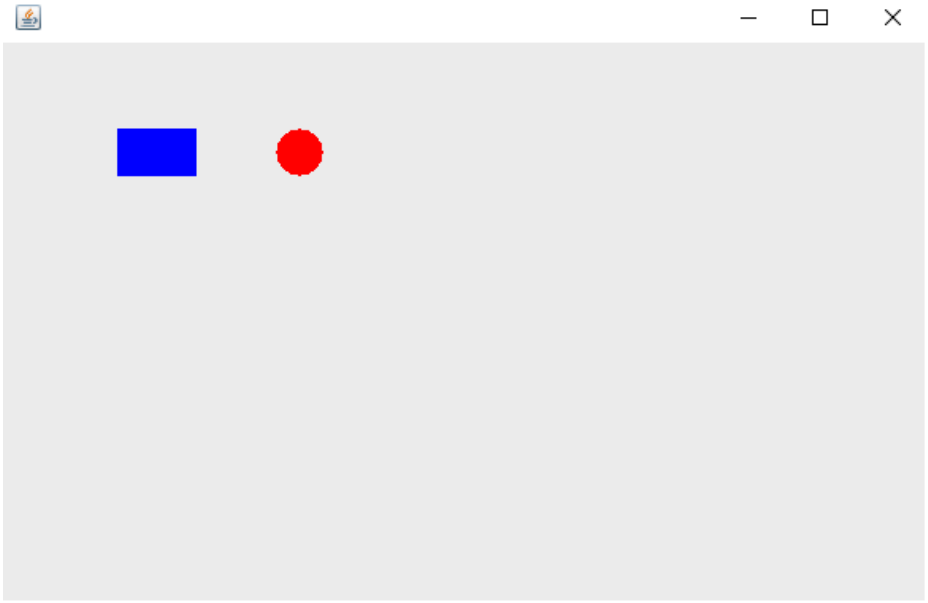
```
    }

    JLabel name, age, occupation;
    JButton fatherDetails, motherDetails;

    @Override
    public void actionPerformed(ActionEvent e) {
        if (e.getSource() == fatherDetails) {
            name.setText("John Doe");
            age.setText("30");
            occupation.setText("Software Engineer");
        } else {
            name.setText("Jane Doe");
            age.setText("25");
            occupation.setText("Network Engineer");
        }
    }
}
```

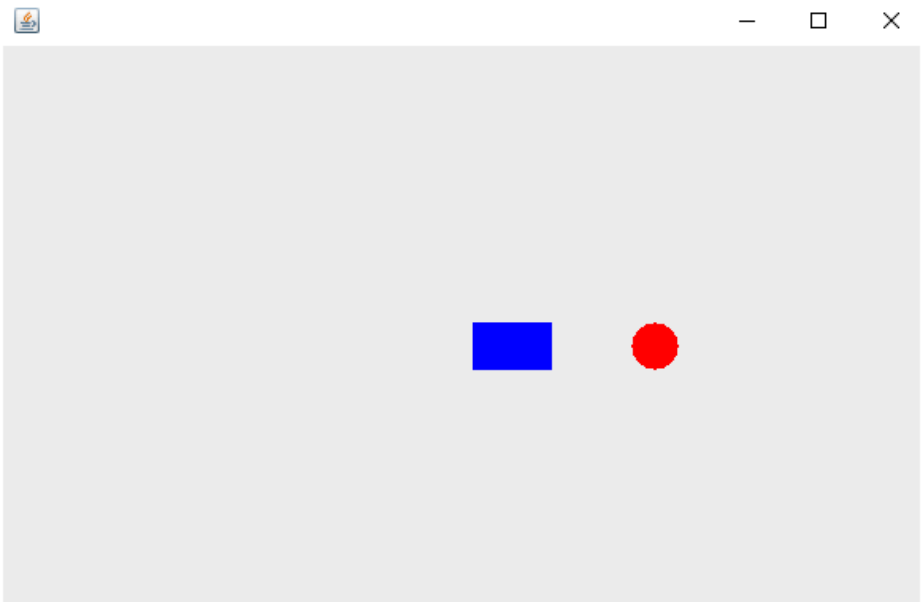


19. Write a program to move different shapes according to the arrow key pressed.

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class ShapeMovement extends JFrame implements KeyListener {

    public ShapeMovement() {
        setSize(600, 400);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setVisible(true);
        addKeyListener(this);
    }

    public void paint(Graphics g) {
        super.paint(g);

        g.setColor(Color.BLUE);
        g.fillRect(x, y, 50, 30);

        g.setColor(Color.RED);
        g.fillOval(x + 100, y, 30, 30);
    }

   public void keyPressed(KeyEvent e) {
        int keyCode = e.getKeyCode();
        switch (keyCode) {
            case KeyEvent.VK_UP:
                y -= dy;
                break;
            case KeyEvent.VK_DOWN:
                y += dy;
                break;
            case KeyEvent.VK_LEFT:
                x -= dx;
                break;
            case KeyEvent.VK_RIGHT:
                x += dx;
                break;
        }
```

```
        repaint(); // Redraw the shapes at the new position
    }

    public static void main(String args[]) {
        new ShapeMovement().setVisible(true);
    }

    private int x = 100; // Initial x-coordinate for shapes
    private int y = 100; // Initial y-coordinate for shapes
    private int dx = 5; // Incremental change in x-coordinate
    private int dy = 5; // Incremental change in y-coordinate

    @Override
    public void keyTyped(KeyEvent ke) {
    }

    @Override
    public void keyReleased(KeyEvent ke) {
    }

}
```

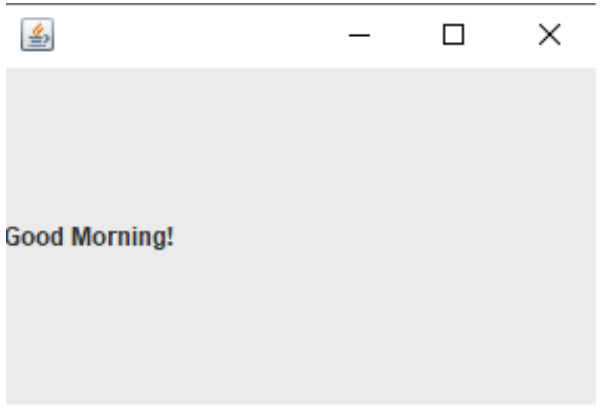


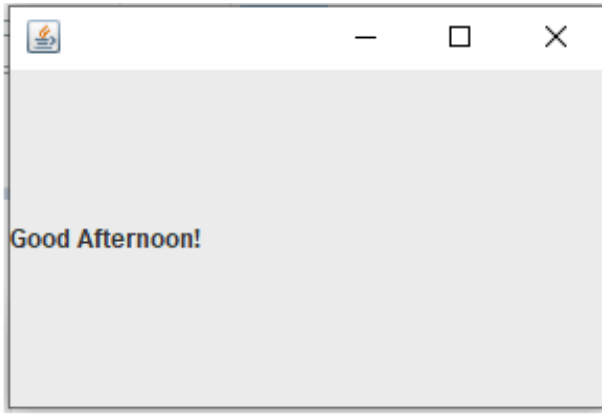


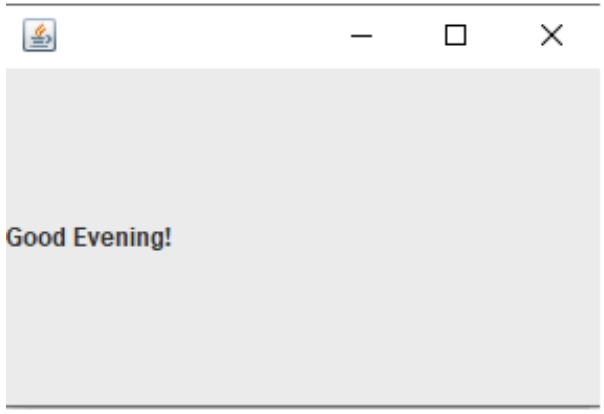
20. Write a java Program to create a window when we press M or m the window displays Good Morning, A or a the window displays Good After Noon E or e the window displays Good Evening, N or n the window displays Good Night

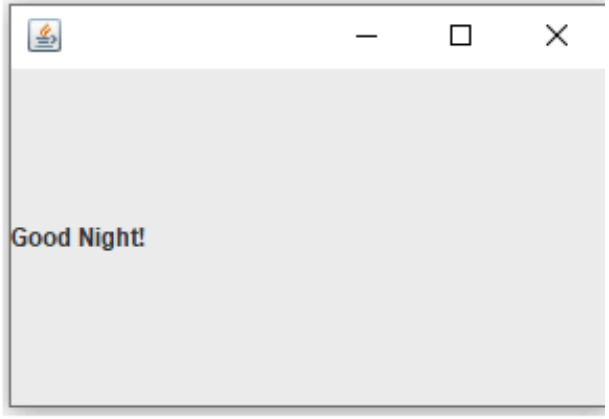
```
import javax.swing.*;
import java.awt.event.*;

public class GreetingWindow extends JFrame implements KeyListener {

    public GreetingWindow() {
        setSize(300, 200);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        text = new JLabel();
        add(text);
        addKeyListener(this);
    }

    @Override
    public void keyTyped(KeyEvent e) {
    }

    @Override
    public void keyPressed(KeyEvent e) {
        char key = e.getKeyChar();
        String message = "";
        switch (key) {
            case 'M':
            case 'm':
                message = "Good Morning!";
                break;
            case 'A':
            case 'a':
                message = "Good Afternoon!";
                break;
            case 'E':
            case 'e':
                message = "Good Evening!";
                break;
            case 'N':
            case 'n':
                message = "Good Night!";
```

```
                break;
            }
            text.setText(message);
        }

        @Override
        public void keyReleased(KeyEvent e) {
        }

        public static void main(String[] args) {
            new GreetingWindow().setVisible(true);
        }
        JLabel text;
}
```





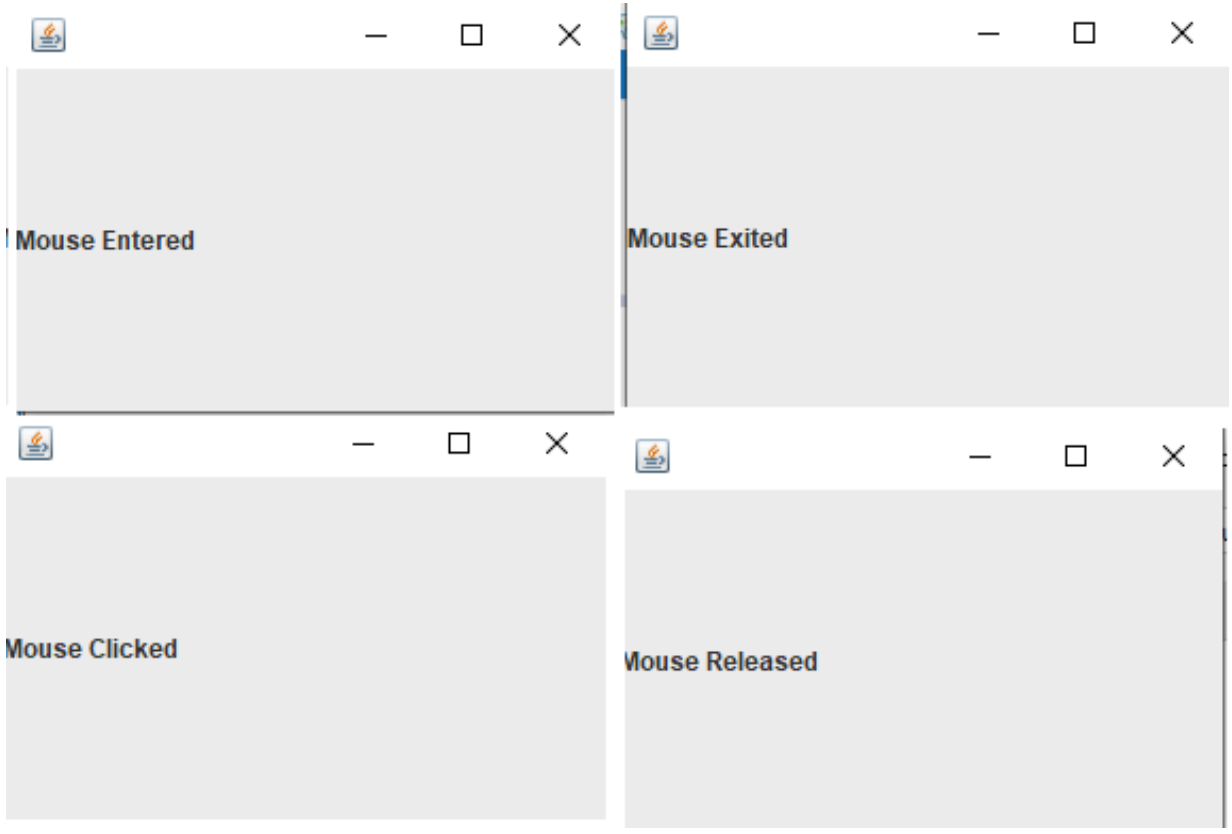




21. Demonstrate the various mouse handling events using suitable examples.

```
import java.awt.event.*;
import javax.swing.*;

public class MouseActions extends JFrame implements MouseListener {

    public MouseActions() {
        lab = new JLabel();
        add(lab);
        setSize(300, 200);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        addMouseListener(this);
    }

    public static void main(String args[]) {
            new MouseActions().setVisible(true);
    }

    @Override
    public void mouseClicked(MouseEvent e) {
        lab.setText("Mouse Clicked");
    }

    @Override
    public void mousePressed(MouseEvent e) {
        lab.setText("Mouse Pressed");
    }

    @Override
    public void mouseReleased(MouseEvent e) {
        lab.setText("Mouse Released");
    }

    @Override
    public void mouseEntered(MouseEvent e) {
        lab.setText("Mouse Entered");
    }

    @Override
    public void mouseExited(MouseEvent e) {
```

```
        lab.setText("Mouse Exited");
    }

    JLabel lab;

}
```



22. Write a program to create menu bar and pull-down menus.

```
import javax.swing.*;

public class MenuFrame extends JFrame {

    public MenuFrame() {
        setSize(300, 200);
        jMenuBar1 = new JMenuBar();
        File = new JMenu();
        New = new JMenuItem();
        Open = new JMenuItem();
        Exit = new JMenuItem();
        Edit = new JMenu();
        Cut = new JMenuItem();
        Copy = new JMenuItem();
        Paste = new JMenuItem();

        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        File.setText("File");

        New.setText("New");
        File.add(New);

        Open.setText("Open");
        File.add(Open);

        Exit.setText("Exit");
        File.add(Exit);

        jMenuBar1.add(File);

        Edit.setText("Edit");

        Cut.setText("Cut");
        Edit.add(Cut);

        Copy.setText("Copy");
        Edit.add(Copy);
```

```
        Paste.setText("Paste");
        Edit.add(Paste);

        jMenuBar1.add(Edit);

        setJMenuBar(jMenuBar1);
    }

    public static void main(String args[]) {
            new MenuFrame().setVisible(true);
    }

    private javax.swing.JMenuItem Copy, Cut, Exit, New, Open, Paste;
    private javax.swing.JMenu File, Edit ;
    private javax.swing.JMenuBar jMenuBar1;
}
```