Software Engineering Lab (CS29006)

Assignment 3: Java Programming

Time: 2:00 pm - 4:55 pm **Date : 06/02/2019**

Instructions for submission

- Give meaningful comments to explain the functionality of each class and function used in your program.
- Make a zip file with and give the name of the zip file as **A3_ < YourRollNo >**.
- The zip file should contain codes for the practice problems and the assignments.
- Submit the zip file to the Moodle system.
- Submit your solution latest by **4:55 pm** on **06/02/2019**.

---

1 Problems for Practice

1.1 Applets

1.1.1 Creating a Basic Applet

Create a **BasicApplet.java** file with the following code.

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

public class BasicApplet extends Applet {
    public void paint(Graphics g) {
        g.drawRect(50, 80, 200, 150);
        g.fillRect(50, 380, 200, 150);
        g.drawOval(350, 80, 200, 150);
        g.fillOval(350, 380, 200, 150);
        g.setColor(Color.red);
        g.drawString("Hello", 20, 20);
    }
}
```

Create a **BasicAppletHtml.html** file with the following code.

```html
<html>
    <head>
        <title>This is a basic applet</title>
    </head>
    <body>
        <applet code="BasicApplet" width=300 height=80></applet>
    </body>
</html>
```

Compile and execute the code using the following commands.
javac BasicApplet.java
appletviewer BasicAppletHtml.html

1.1.2 Drawing a Human Face

Create a **Face.java** file with the following code.

```
import java.awt.*;
import java.applet.*;
public class Face extends Applet
{
    public void paint(Graphics g)
    {
        g.drawOval(40, 40, 120, 150);
        g.drawOval(57, 75, 30, 20);
        g.drawOval(110, 75, 30, 20);
        g.fillOval(68, 81, 10, 10);
        g.fillOval(121, 81, 10, 10);
        g.drawOval(85, 100, 30, 30);
        g.fillArc(60, 125, 80, 40, 180, 180);
        g.drawOval(25, 92, 15, 30);
        g.drawOval(160, 92, 15, 30);
    }
}
```

Create a **ShowFace.html** file with the following code.

```
<html>
    <head>
        <title>This is just a face</title>
    </head>
    <body>
        <applet code="Face" width=300 height=80></applet>
    </body>
</html>
```

Compile and execute the code using the following commands.
javac Face.java
appletviewer ShowFace.html
Now modify the above code to include a APPLET tag in the Java file.

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

/*
<applet code="Face" width=200 height=60>
</applet>
*/

public class Face extends Applet
{
    public void paint(Graphics g)
    {
        g.drawOval(40, 40, 120, 150);
        g.drawOval(57, 75, 30, 20);
        g.drawOval(110, 75, 30, 20);
        g.fillOval(68, 81, 10, 10);
        g.fillOval(121, 81, 10, 10);
        g.drawOval(85, 100, 30, 30);
        g.fillArc(60, 125, 80, 40, 180, 180);
        g.drawOval(25, 92, 15, 30);
        g.drawOval(160, 92, 15, 30);
    }
}
```

Compile and execute the code using the following commands.
javac Face.java
appletviewer Face.java

1.2 Swing

1.2.1 Label, Radio Button, and Button

Type the following code and check the output.

```
import javax.swing.*;

public class SwingRadioButton {
    JFrame f;

    SwingRadioButton() {
        f = new JFrame();
        JLabel l1=new JLabel("Select your meal preference:");
        JRadioButton r1 = new JRadioButton("A) Veg");
        JRadioButton r2 = new JRadioButton("B) Non-Veg");
        JButton b=new JButton("Submit");
        l1.setBounds(75,30, 200,30);
        r1.setBounds(75, 80, 100, 30);
        r2.setBounds(75, 100, 100, 30);
        b.setBounds(75,150,100,30);
        ButtonGroup bg = new ButtonGroup();
        bg.add(r1);
        bg.add(r2);
        f.add(l1);
        f.add(r1);
        f.add(r2);
        f.add(b);
        f.setSize(300, 300);
        f.setLayout(null);
        f.setVisible(true);
    }

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

1.2.2 Option Pane

Type the following code and check the output.

```
import javax.swing.*;

public class SwingOptionPane {
    JFrame f;

    SwingOptionPane() {
        f = new JFrame();
        JOptionPane.showMessageDialog(f, "Hello, Welcome to Swing");
    }

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

1.2.3 Text Area with Action Listener

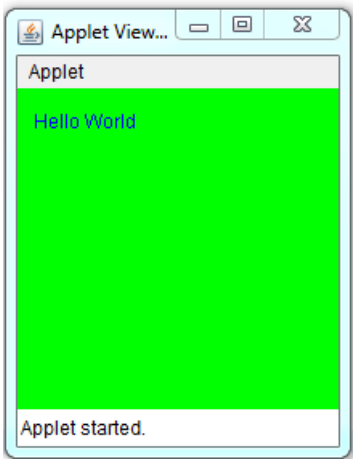
Type the following code and check the output.

```
1  import javax.swing.*;
2  import java.awt.event.*;
3  public class TextAreaExample implements ActionListener {
4      JLabel l1, l2;
5      JTextArea area;
6      JButton b;
7
8      TextAreaExample() {
9          JFrame f = new JFrame();
10         l1 = new JLabel();
11         l1.setBounds(50, 25, 100, 30);
12         l2 = new JLabel();
13         l2.setBounds(160, 25, 100, 30);
14         area = new JTextArea();
15         area.setBounds(20, 75, 250, 200);
16         b = new JButton("Count Words");
17         b.setBounds(100, 300, 120, 30);
18         b.addActionListener(this);
19         f.add(l1);
20         f.add(l2);
21         f.add(area);
22         f.add(b);
23         f.setSize(450, 450);
24         f.setLayout(null);
25         f.setVisible(true);
26     }
27     public void actionPerformed(ActionEvent e) {
28         String text = area.getText();
29         String words[] = text.split("\\s");
30         l1.setText("Words: " + words.length);
31         l2.setText("Characters: " + text.length());
32     }
33     public static void main(String[] args) {
34         new TextAreaExample();
35     }
36 }
```

2 Assignments

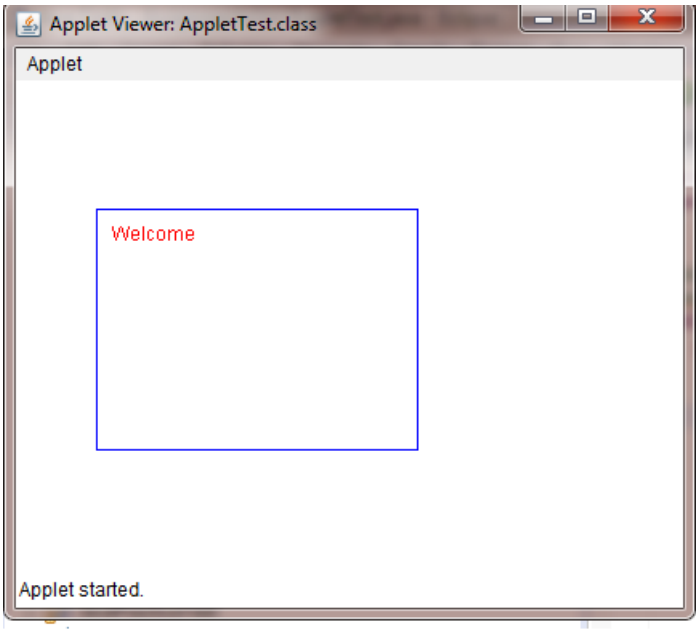
1. Write an applet to print a string in blue color. Change the background color of the applet window to green.

   Expected Output:



2. Write an applet that draws a rectangle. The rectangle should have a width and height of 200 and 150 pixels respectively. Print a "Welcome" message that is fully contained inside the rectangle. The border of the rectangle should be blue. The color of the message should be red.

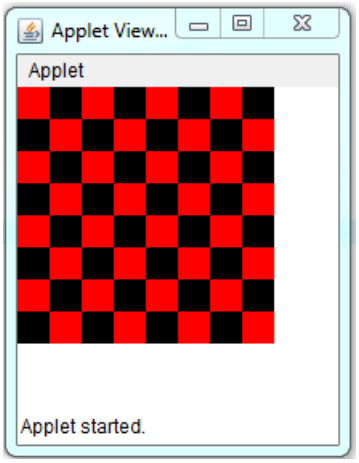
   Expected Output:



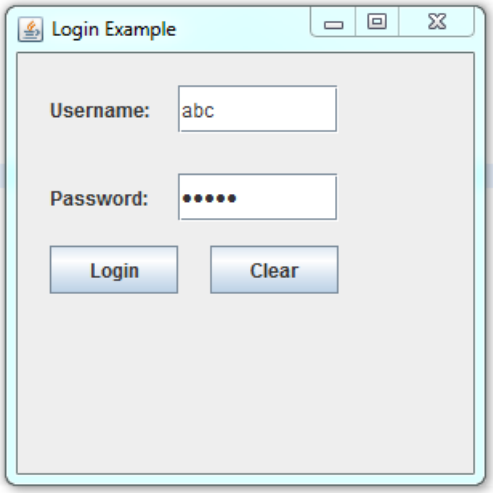
3. Write an applet that draws a checkerboard. Assume that the size of the applet is 160 by 160 pixels. Each square in the checkerboard is 20 by 20 pixels. The checkerboard contains 8 rows of squares and 8 columns. The squares are red and black.

   Hint: Here is a tricky way to determine whether a given square is red or black: If the row number and the column number are either both even or both odd, then the square is red. Otherwise, it is black. Note that a square is just a rectangle in which the height is equal to the width, so you can use *fillRect()* to draw the squares.

   Expected Output:



4. Write a Java Swing code to create the following GUI.



5. Write a Java Swing code to create the following GUI.