

4. GUI Programming - Draw the GUI that is generated by the following code. (1.0%)

Assume the image usc . jpg is



Assume the image ucla . jpg is



```
import java.awt.BorderLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.Box;
import javax.swing.BoxLayout;
import javax.swing.ButtonGroup;
import javax.swing.ImageIcon;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JRadioButton;

public class Problem5 extends JFrame {
    private JLabel jl, jl1, jl2;
    private ImageIcon ii, iil;
    public Problem5() {
        super("Problem 5");
        setSize(200, 300);
        setLocation(500, 300);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        JPanel jp = new JPanel();
        jp.setLayout(new BoxLayout(jp, BoxLayout.X_AXIS));
        ii = new ImageIcon("usc.jpg");
        iil = new ImageIcon("ucla.jpg");
        jl = new JLabel(ii);
        jp.add(Box.createGlue());
        jp.add(jl);
        jl1 = new JLabel("USC");
        jp.add(jl1);
        jp.add(Box.createGlue());
        jl2 = new JLabel(iil);
        jp.add(jl2);
        add(jp, BorderLayout.CENTER);

        JRadioButton jrb = new JRadioButton("USC", true);
        JRadioButton jrb1 = new JRadioButton("UCLA", false);
        ButtonGroup bg = new ButtonGroup();
        bg.add(jrb);
        bg.add(jrb1);
        JPanel jpl = new JPanel();
        jpl.add(jrb);
        jpl.add(jrb1);
        add(jpl, BorderLayout.NORTH);
        setVisible(true);
    }
    public static void main(String [] args) {
        new Problem5();
    }
}
```

5. **Inheritance** – There are two errors in the following code. Describe the errors and modify the code to fix them. (2.0%):

```
1 interface C {
2     public int m();
3     public abstract int n();
4 }
5
6 abstract class D implements C {
7     public int m() {
8         return 3;
9     }
10 }
11
12 abstract class E extends D {
13     public int n() {
14         return 4;
15     }
16     public int o() {
17         return 5;
18     }
19 }
20
21 class F extends E {
22
23 }
24
25 public class Problem3 {
26     public static void main(String [] args) {
27         F f = new F();
28         f.m();
29         f.n();
30         f.o();
31         E e = new E();
32         e.m();
33         e.n();
34         e.o();
35         C c = new F();
36         c.m();
37         c.n();
38         c.o();
39     }
40 }
```

6. Inner Classes – `KeyListener` is an interface that contains three methods – `keyPressed`, `keyTyped`, and `keyReleased`, all three of which take a `KeyEvent` as a parameter. Write the code using an anonymous inner class to add a `KeyListener` to a `JTextField` that prints out “Hello” to the command line when a user releases the ‘h’ key on the keyboard. **(1.5%)**

7. Garbage Collection – The garbage collector provides one of the biggest advantages to Java over C++, though many Java developers inaccurately believe that a Java program cannot crash due to running out of memory. Write a snippet of code that would cause a Java program to throw an `OutOfMemoryError`. **(1.0%)**

8. **Software Engineering** – Scrum is arguably the most popular software engineering methodology in use right now. Give two reasons why you think that is the case. **(0.5% + 0.5%)**

9. **Anonymous Inner Classes** – The following code may appear to some people to be instantiating an interface, which we learned is not possible in Java. If the following code does not compile, explain why and fix it. If the following code does compile, explain what is happening on lines 7-14. **(1.0%)**

```
1  public class Problem6 {
2      public Problem6(I i) {
3          i.a();
4          i.b();
5      }
6      public static void main(String [] args) {
7          new Problem6(new I() {
8              public void a(float f) {
9                  System.out.println("a");
10             }
11             public void b() {
12                 System.out.println("b");
13             }
14         });
15     }
16 }
17 interface I {
18     void a(int i);
19     void b();
20 }
```