

**CSCI 201L Midterm – Written SOLUTION**  
**Fall 2014**  
**12% of course grade**

1. **Classes/Objects** – Why is it good practice to make all member variables private? What is the potential problem if some member variables are not private? **(1.5%)**

**1.0% Good object-oriented principles keep the scope of variables as narrow as possible so other code is not able to modify the values of variables without the class who owns them knowing.**

**0.5% When debugging, if there is a problem with the value of a variable, you only need to look in the current class if the variables are private.**

2. **Interfaces** – What is the purpose of an interface in Java? Why does Java have interfaces instead of just abstract classes with all of the methods abstract? **(1.5%)**

**0.5% Interfaces are Java’s solution to only having single inheritance, though you can inherit from more than one object through interfaces.**

**0.5% Abstract classes still inherit from the Object class (or possibly other classes), and all of the methods in those classes are not abstract.**

**0.5% Abstract classes are still limited by single inheritance. A class will only be able to inherit from one abstract class instead of multiple interfaces, though the programmer’s desired functionality will probably be similar.**

3. **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 }
```

0.5% E is an abstract class and therefore can't be instantiated.  
Fix (0.5%)  
The line could be changed to:  
E e = new F();

0.5% The method o is undefined for class C. The only methods that can be called on an object are those which are defined in the compile-time type of the object.  
Fixes (0.5% for one of the following)

1. We could add an abstract method o into interface C.
2. We could change line 35 to say E e = new F();

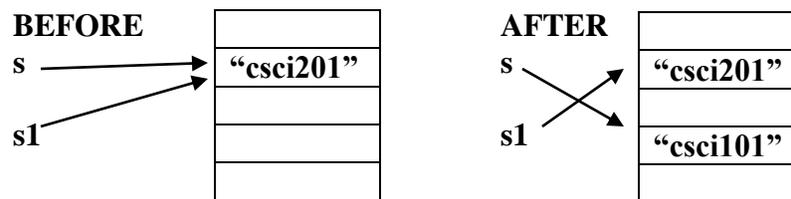
4. **Strings** – Strings are immutable in Java. What does immutable mean? Provide an example with two Strings and a main memory diagram that shows how immutability works. (1.0%)

**0.5% Immutable means that the String will not change after being initially created. The reference of the variable may change, but the String will not.**

**0.5% for an example similar to the following**

```
String s = "csci201";  
String s1 = s;  
s.replace('2', '1');  
System.out.println(s); // csci101  
System.out.println(s2); // csci201
```

The String s2 will still print “csci201” because the value of the String originally pointed to by s will not change. The reference of s will change and point to a new location in memory with the new String.



5. **File I/O** – Give two conditions that would cause an IOException to be thrown when trying to write to a file. (1.0%)

**0.5% for each reason, with a maximum of 1.0%**

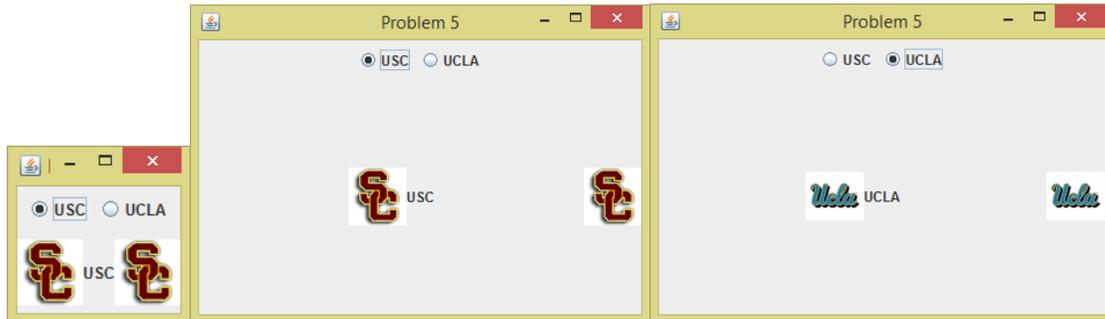
**If you don't have permission to write to a file.**

**If you try to write to a directory instead of a file.**

**If you don't have permission to create a file and the file doesn't exist.**

**If the file can't be opened for writing.**

6. **GUI** - Write the code to generate the following GUI. The screenshots are all from the same execution of the program with the window resized, which should help you to determine the layout manager(s) in use. Assume you have access to an image named `usc.jpg` and `ucla.jpg` in your project directory. **(2.5%)** Add an event to the radio button that changes the bottom part when clicked, as shown in the screenshots. **(2.5%)**



**2 correct versions of the code are shown on the following two pages.**

**Layout Managers (1.5%)**

**0.5% if layout manager of frame is BorderLayout, which is the default**

**One of the following:**

**1.0% if a JLabel is used with an image and text in the center and a JLabel for the image in the east**

**OR**

**0.5% if a BorderLayout is used inside a JPanel for the center of the BorderLayout**

**0.5% if glue is used in the BorderLayout**

**If they used a different arrangement of layout managers that seems to work, give 0.5%**

**Components (1.0%)**

**0.3% if they have JRadioButtons**

**0.3% if they have the radio buttons in a ButtonGroup**

**0.3% if they have Image Icons**

**0.1% if they have a JLabel for the USC text**

**Event Handling (2.0%)**

**0.5% if they have an ActionListener for the USC radio button**

**0.5% if they have an ActionListener for the UCLA radio button**

**0.5% if they set the icons to be the different image in each ActionListener**

**0.5% if they set the text to be different in each ActionListener**

**Overall (0.5%)**

**0.2% if they have a class inheriting from JFrame**

**0.1% setting the title through super() or setTitle()**

**0.1% setting the size of the JFrame**

**0.1% setting the JFrame visible**

**Anything else that is included or omitted should not be graded. Only grade based on the above criteria.**

```

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(ii);
        jp.add(jl2);
        add(jp, BorderLayout.CENTER);

        JRadioButton jrb = new JRadioButton("USC", true);
        jrb.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent ae) {
                jl.setIcon(ii);
                jl1.setText("USC");
                jl2.setIcon(ii);
            }
        });
        JRadioButton jrb1 = new JRadioButton("UCLA", false);
        jrb1.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent ae) {
                jl.setIcon(iil);
                jl1.setText("UCLA");
                jl2.setIcon(iil);
            }
        });
        ButtonGroup bg = new ButtonGroup();
        bg.add(jrb);
        bg.add(jrb1);
        JPanel jp1 = new JPanel();
        jp1.add(jrb);
        jp1.add(jrb1);
        add(jp1, BorderLayout.NORTH);
        setVisible(true);
    }
    public static void main(String [] args) {
        new Problem5();
    }
}

```

```

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

@SuppressWarnings("serial")
public class MidtermFrame extends JFrame implements ActionListener {
    JRadioButton jr1, jr2;
    JLabel jl1, jl2;
    ImageIcon uscimg, uclaimg;

    public MidtermFrame() {
        setLayout( new BorderLayout() );

        JPanel topPanel = new JPanel(); //panel for radio buttons
        add( topPanel, BorderLayout.NORTH );

        jr1 = new JRadioButton( "USC" );
        jr2 = new JRadioButton( "UCLA" );

        ButtonGroup bg = new ButtonGroup();
        bg.add( jr1 );
        bg.add( jr2 );

        topPanel.add( jr1 );
        topPanel.add( jr2 );

        uscimg = new ImageIcon( "usc.jpg" );
        uclaimg = new ImageIcon( "ucla.jpg" );

        jl1 = new JLabel( "USC" );
        jl1.setIcon( uscimg );
        jl1.setHorizontalAlignment( JLabel.CENTER );
        jl1.setVerticalAlignment( JLabel.CENTER );
        add( jl1, BorderLayout.CENTER );

        jl2 = new JLabel( uscimg );
        jl2.setHorizontalAlignment( JLabel.CENTER );
        jl2.setVerticalAlignment( JLabel.CENTER );
        add( jl2, BorderLayout.EAST );

        jr1.addActionListener( this );
        jr2.addActionListener( this );

        setSize( 200, 300 );
        setDefaultCloseOperation( JFrame.EXIT_ON_CLOSE );
        setVisible( true );
    }

    public void actionPerformed( ActionEvent ae ) {
        if ( ae.getSource() == jr1 ) {
            jl1.setIcon( uscimg );
            jl1.setText( "USC" );
            jl2.setIcon( uscimg );
        } else {
            System.out.println("in else");
            jl1.setIcon( uclaimg );
            jl1.setText( "UCLA" );
            jl2.setIcon( uclaimg );
        }
    }

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

```