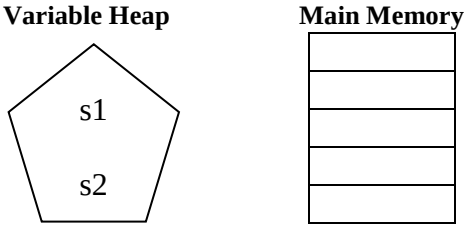
CSCI 201L Midterm – Written
Spring 2015
10% of course grade

1. **Strings** – Do the following two blocks of code produce the same results? Fill in the main memory diagram *and* explain. **(0.5% + 0.5%)**

```
public boolean equalStrings(String s1, String s2) {
   return s1 == s2;
}

public boolean sameStrings(String s1, String s2) {
   return s1.equals(s2);
}
```



2. **Serialization** – We talked about using serialization with file I/O. Give two advantages that serializing an object has over writing code to save all of the variables in a class to a file. **(0.75% + 0.75%)**

3. **Exception Handling** – Does the following code compile? If it does, what is the output? If it does not, explain the error and provide a solution. **(0.5% + 0.5%):**

```
1   public class Problem3 {
2     Problem3(int i1, int i2) throws M {
3       try {
4         if (i1 < i2) {
5           throw new M();
6         }
7       } catch (NumberFormatException nfe) {
8         System.out.println("nfe: " + nfe.getMessage());
9       } finally {
10        System.out.println("leaving constructor");
11      }
12    }
13
14    public static void main(String [] args) {
15      new Problem3(8, 10);
16    }
17
18    private class M extends Exception {
19      public String getMessage() {
20        return "Numbers not formatted correctly";
21      }
22    }
23  }
```

4. **Inheritance –** Provide a code snippet to exhibit the principle of proximity precedence. Explain the principle with respect to your example. **(0.5% + 0.5%)**

5. **Generics** – Before Java added generics into the language, there were errors that could only be found at runtime. Explain what type of errors generics allow us to find at compile time that would only be caught at runtime in a language that does not have generics. Provide a code snippet to help explain your answer. **(0.5% + 1.0%)**

6. **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. **(0.5% + 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() {
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();
19    void b();
20  }
```

7. Looking at the following code, draw out the GUI that is shown when the application first runs. The code below does compile and run, so do not try to find any errors. **(2.5%)**

```
1  import java.awt.*;    // note: imported * to save space for exam
2  import javax.swing.*; // not recommended coding practice
3  public class Problem7 extends JFrame {
4    public Problem7() {
5      super("Problem 7");
6      setSize(400, 200);
7      setLocation(100, 100);
8      setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
9      createGUI();
10   }
11   private void createGUI() {
12     JTabbedPane jtp = new JTabbedPane();
13     JPanel jp1 = new JPanel();
14     jp1.setLayout(new GridLayout(1, 2));
15     jp1.add(new JLabel("world"));
16     JPanel jp2 = new JPanel();
17     JPanel jp3 = new JPanel();
18     jp3.setLayout(new GridBagLayout());
19     GridBagConstraints gbc = new GridBagConstraints();
20     gbc.gridx = 0;
21     gbc.gridy = 0;
22     jp3.add(new JLabel("mercury"), gbc);
23     gbc.gridx = 1;
24     jp3.add(new JComboBox(new String[]{"venus", "earth"}), gbc);
25     gbc.gridx = 0;
26     gbc.gridy = 1;
27     gbc.gridwidth = 2;
28     jp3.add(new JScrollPane(new JTextArea("mars\njupiter\nsaturn", 3,
                                          20)), gbc);
29     gbc.gridx = 2;
30     gbc.gridy = 0;
31     gbc.gridwidth = 1;
32     gbc.gridheight = 3;
33     gbc.anchor = gbc.FIRST_LINE_START;
34     jp3.add(new JTree(new String[]{"uranus", "neptune", "pluto?"}), gbc);
35     gbc.gridx = 0;
36     gbc.gridy = 2;
37     gbc.gridwidth = 3;
38     gbc.gridheight = 1;
39     gbc.ipadx = 100;
40     gbc.anchor = gbc.CENTER;
41     jp3.add(new JButton("Milky Way Planets"), gbc);
42     jtp.add("1", jp1);
43     jtp.add("2", jp2);
44     jtp.add("3", jp3);
45     jtp.setSelectedComponent(jp3);
46     add(jtp, BorderLayout.CENTER);
47   }
48   public static void main(String [] args) {
49     Problem7 p7 = new Problem7();
50     p7.setVisible(true);
51   }
52 } // end class Problem7
```