1. **Inheritance** – Answer the following questions about inheritance.
   
   a. Does Java allow overloading, overriding, and redefining of methods? (0.5%)
   
   b. Explain what is meant by **overloading**. Provide a code snippet as an example. (0.5%)
   
   c. Explain what is meant by **overriding**. Provide a code snippet as an example. (0.5%)
   
   d. Explain what is meant by **redefining**. Provide a code snippet as an example. (0.5%)
2. **Exception Handling** – Java, unlike C++, has both checked and unchecked exceptions. Explain why unchecked exceptions in Java do not need to be handled by a corresponding `try` block. (0.5%)

3. **User Interface Design** – Explain two differences between JDialog and JFrame. (0.5%+0.5%)
4. Generics – Answer the following questions based on the generic code below.

```java
public class Problem4 {
    public static<T, T1 extends T> void meth(T1 [] list) {
        T num = list[0];
        System.out.println(num);
    }

    public static void main(String[] args) {
        Integer[] integers = {1, 2, 3, 4, 5};
        String[] strings = {"London", "Paris", "New York"};

        Problem4.<Number, Integer>meth(integers);
        Problem4.<Object, Object>meth(strings);
    }
}
```

a. Does the code throw a compile-time error, have a warning, throw a runtime error, or compile and run? (0.5%) 

Answer either question (b) or (c) based on your answer to (a) above. If you answer both questions (b) and (c), you will not get credit for either one.

b. If the code throws a compile-time error, has a warning, or throws a runtime error, explain the problem and provide a solution to it in the above code. (0.5%) 

c. If the code compiles and runs, explain what is happening on lines 11, 12, 2, and 3. (0.5%)
5. **Layout Managers** – For each layout manager below, explain its functionality. Include whether the preferred height and width of a component is acknowledged, how components are laid out when added, and what happens to the components when the frame is resized. Draw (no code needed) a typical GUI that uses each layout manager. (0.5% + 0.5% + 0.5%)

   a. **BoxLayout (Y_AXIS)**

   b. **GridLayout**

   c. **GroupLayout**
6. **Inner Classes** – Your friend from UCLA sees some of your GUI code and tells you that it won’t compile because you have instantiated an abstract class. You respond by saying that this is an anonymous inner class. Explain to your less-educated UCLA friend what is actually happening with an anonymous inner class. Use the following code as an example. (1.0%)
7. **Option Panes and Dialogs** – Give two differences and two similarities between option panes and dialog boxes in Java. (0.25% + 0.25% + 0.25% + 0.25%)

*Similarity #1 –*

*Similarity #2 –*

*Difference #1 –*

*Difference #2 –*
8. **Graphics** – Draw the GUI rendered by the following code. (1.0%)
9. **Tables and Trees** – Draw the GUI that is generated by the following code. (1.0%)