433 Practice Final Questions - COM, JavaBeans, Design Patterns

The final exam is cumulative, covering topics from the entire semester.
These questions mainly cover topics since the 2nd midterm. Look at the two midterms and their practice exams for questions on topics covered earlier in the course.

1. In C++, COM and Java, compare how it is determined that objects are no longer in use and their resources can be recycled. How can a programmer specify actions to be taken when an object is recycled? Try to find something nice to say about all three.

2. Compare, briefly, the relative advantages of using RMI or sockets for a distributed computation.

3. For the function `Ticket register(Person p)`, the following are possible pre and post conditions (t is used to refer to the return value).

   (a) Pre: p is not already registered,
   Post: p is registered and t is a Ticket for p

   (b) Pre: p is not already registered,
   Post: (returns normally, p is registered and t is a Ticket for p)
   or (throws SoldOutException) Exception)

   (c) Pre: true,
   Post: (returns normally, p is registered and t is a FirstClassTicket for p)
   or (throws AlreadyRegisteredException) Ticket for p) AlreadyRegisteredException) SoldOutException)

   (d) Pre: Event not sold out,
   Post: (returns normally, p is registered and t is a Ticket for p)
   or (throws AlreadyRegisteredException)

   Assume that $\texttt{f}_a$ returns to `f` with pre and post conditions given in a, etc. Note that FirstClassTicket is a subtype of Ticket, and that AlreadyRegisteredException and SoldOutException do not have a subtype relationship.

   - Show a table that describes whether or not it is legal to substitute a function with the described pre/post conditions for each of the other functions. In other words, which pairs of pre/post conditions could represent legal relationships between a client function's expectations/requirements and the function's interface contract? (The same pairs should also represent legal relationships between the interface contract and implementation guarantees made by the function provider)

4. (short answers - no more than 3 sentences)

   (a) What is the QueryInterface function, and how does it typically work?

   (b) What are the security implications of declaring that a Java class implements Serializable?

   (c) How is Leasing used in distributed systems, and why is it useful?

   (d) Give an example where the Singleton design pattern would be useful.

5. Write a JavaBean that implements a Rectangle interface, with properties of the following types:

   - Color color;
   - Point upperLeft;
   - Point bottomRight;
   - boolean visible;

   where Point has interface:

```
interface Point {
  int getX();
  int getY();
  void setX(int);
  void setY(int);
}
```

Include functions that allow other Beans to register as listeners on bound properties, and make the color and visible properties bound so that they trigger events in listeners.

6. Fill in code where indicated so that the visitAll method properly applies the visitor v to all nodes in the set s. Note that the additional code should not use instanceof, casts or reflection.

```
interface Visitor {
        visitA(A a); // This method should be called on all A nodes
        visitB(B b); // This method should be called on all B nodes
        }

abstract Class Node {
        static void visitAll(Set s, Visitor v) {
                for(Iterator i = s.iterator(); i.hasNext(); ) {
                        Node n = (Node) i.next();
                        // fill in here
                        }
                }
        // fill in here
        }

Class A extends Node {
        // fill in here
        }
Class B extends Node {
        // fill in here
        }
```