

433 Practice 2nd Midterm

- Define a Java class `Room`. When a room is created, you specify the capacity (an integer). Implement two functions, `enter()` and `leave()`. Keep track of the number of threads that have entered a room. If the room is at capacity, a new thread trying to enter the room will block until somebody leaves and the thread can enter without pushing the room over capacity.
Don't worry about handling recursive locks.
 - Provide an additional function `doInRoom(Runnable r)` that enters the room, invokes the `run` method of `r`, and then leaves the room. The `leave()` method is invoked even if the `run` method of `r` throws an exception (although execution of `doInRoom` still terminates with the exception thrown by the `run` method).

- Say a server wanted to be able to send a stream of messages to a client. You want to make sure that the client receives the messages in order, without skipping any. Allow for the machine the client is running on to crash, with the client migrating to another computer.

Describe the high level design of a system to handle these goals.

- Why does a remote object need to implement a remote interface? Alternatively, what is special about a remote interface?
- Say you had implemented a `HashMap` class (ignore the one already provided in the JDK). Is it reasonable for you to declare that your class is `Serializable`? What would the default serialization do to a `HashMap`? What if some values appeared multiple times?

If you wrote a custom serialization, how might you do it differently to make serialization more efficient?

- Give an example where the Singleton design pattern would be useful.
- Using the Abstract Factory design pattern, show the declarations of Java classes that are widget factories for the Windows and X operating environments to instantiate GUI objects, with the following interface:

```
interface WidgetFactory {
    public Window createWindow();
    public Icon createIcon(String filename);
    public Menu createMenu(String name, String[] entries);
}
```

Also provide code for a client that uses the X widget factory to create an icon from a file named *icon.gif* and a menu named *Window Ops* with two entries, *Move Window* and *Resize Window*.

What restrictions/requirements are there on the types of the objects returned from the methods in the Windows and X widget factories?