

433 Quiz Answer, January 30th

1. (50 points) Write appropriate operator=, copy constructor, and destructor methods for the following class.

```
class SimpleArray {
    int length;
    int * data;
public:
    SimpleArray(int len) {
        length = len;
        data = new int[len];
    }
    SimpleArray(const SimpleArray & that) {
        length = that.length;
        data = new int[length];
        for(int i = 0; i < length; i++)
            data[i] = that.data[i];
    }
    SimpleArray& operator=(const SimpleArray & that) {
        if (this == &that) return *this;
        delete [] data;
        length = that.length;
        data = new int[length];
        for(int i = 0; i < length; i++)
            data[i] = that.data[i];
        return *this;
    }
    ~SimpleArray() {
        delete [] data;
    }
    int get(int idx) const {
        return data[idx];
    }
    void set(int idx, int val) {
        data[idx] = val;
    }
    int getLength() const {
        return length;
    }
};
```

2. (50 points) Java synchronization

Design a simple one-place buffer for use a multithreaded Java applications. The buffer can hold one object; when the buffer is allocated, it is empty. If a thread tries to take from an empty buffer, the thread blocks until it can return something. If a thread tries to put into a full buffer, the thread blocks until the buffer has been emptied and the request can be fulfilled. Put your answer on the back of the page.

Your class should implement the following interface:

```
package cmsc433;
public interface OnePlaceBuffer {
    public Object take() throws InterruptedException;
    public void put(Object o) throws InterruptedException;
}
```

In my solution, I'm going to assume that take() should never return null, and that doing put(null) is a no-op.

```
public class Buf implements cmsc433.OnePlaceBuffer {
    private Object data;
    public synchronized Object take() throws InterruptedException {
        while (data == null)
            wait();
        Object tmp = data;
        data = null;
        notifyAll();
        return tmp;
    }

    public synchronized void put(Object o) throws InterruptedException {
        while (data != null)
            wait();
        data = o;
        notifyAll();
    }
}
```