Question 5 (10 marks total)

Consider the following two classes related by inheritance:

```
public class Lock {
    private boolean isLocked;

    public Lock(boolean isLocked) {
        this.isLocked = isLocked;
    }

    public boolean isLocked() {
        return this.isLocked;
    }

    protected void lock() {
        this.isLocked = true;
    }

    protected void unlock() {
        this.isLocked = false;
    }
}


public class CombinationLock extends Lock {
    private Combination combo;

    public CombinationLock(Combination combo) {
        [PART A]
        this.combo = new Combination(combo);
    }

    public CombinationLock(CombinationLock other) {
        [PART B]
    }

    @Override
    public void lock() {
        [PART C]
        this.combo.shuffle();   // randomly shuffles the dials on the lock
    }

    public void unlock(Combination tryMe) {
        if (this.combo.equals(tryMe)) {
            [PART D]
        }
    }
}
```

A `Lock` resembles a physical lock and can be in one of two states: locked or unlocked. A `CombinationLock` is a kind of `Lock` that resembles a physical combination lock made up of numbered dials. A `CombinationLock` can be unlocked if it is provided the correct combination.

Part A (2 marks):

What single line of Java code would you write on the line labelled `[PART A]` to complete the constructor so that the `CombinationLock` is locked?

Part B (4 marks):

For the line labelled `[PART B]` you need to complete the copy constructor. Provide two different implementations of the copy constructor:

- one implementation should use constructor chaining
- the second implementation should not use constructor chaining

Part C (2 mark):

The `lock()` method needs to set the state of the lock to locked on the line labelled `[PART C]`. What single line of Java code would you write to do so?

Part D (2 mark):

The `unlock()` method needs to set the state of the lock to unlocked on the line labelled `[PART D]`. What single line of Java would you write to do so?