Note: No model answers provided for these questions but you are welcome to post your answers in the forum and discuss with others.

(a) Sketch a class diagram for the code given below. Follow the layout given. Show association as lines, not as attributes. (estimated time: 8-10 minutes)

```
interface Readable {
    // ...
}
```

```
interface Countable extends Readable {
    // ...
}
```

```
abstract class Document
        implements Readable {
    // ...
}
```

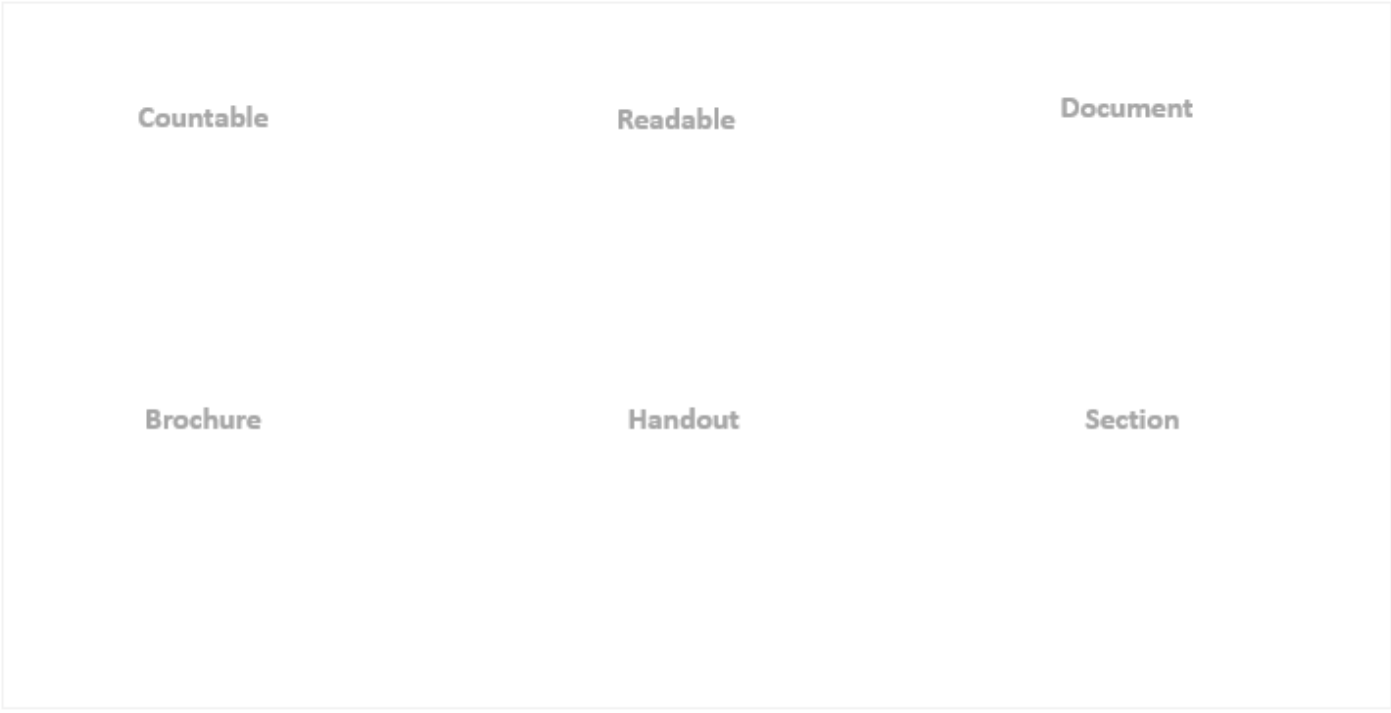
```
class Handout extends Document {
    public List<Brochure> referenced;
    private List<Section> sections;
    protected boolean isFull;
    private int page = 0;
    String heading;
    // ...
}
```

```
class Brochure {
    Brochure closest;
    Handout cover; //cover page
    // ...
}
```

```
class Section {
    public static void capacity(Handout h) {
        // ...
    }
    // ...
}
```

Other info:

- A `Handout` object composed of at least 2 and no more than 5 `Section` objects.
- Each `Brochure` have exactly one `Handout` to serve as the *cover* page.
- A `Handout` can be referenced by the text in any number of `Brochures`.



(b) Sketch an object diagram to match the object structure for one `Handout` object that has the heading "Intro" and two `Section` objects. (estimated time: 2-4 minutes)

(c) Sketch a sequence diagram to illustrate the interactions resulting from `run()` method given below. (estimated time: 8-10 minutes)

```
void run(Factory f, int status) {
    Unit u = f.create();       // line 1
    u.creator().increase(1);   // line 2
    switch (status) {          // line 3
        case 2:
            f.increase(2);
            break;
        case 3:
            f.increase(3);
    }
    f.pad();                   // line 4
}
```

```
class Factory {
    static int count = 0;

    Unit create() {
        return new Unit(this);
    }

    void increase(int amount) {
        count = count + amount;
    }

    void pad() {
        increase(5);
    }
}
```

```
class Unit {
    private Factory factory;

    public Unit(Factory creator) {
        this.factory = creator;
    }

    Factory creator() {
        return factory;
    }
}
```

(d) Sketch an activity diagram to illustrate the workflow of unlocking a door using a biometric device, given below.

1. User places the hand on the reader device.
2. Three parallel processes (`p1, p2, p3`) verify three aspects captured by the device.
3. If all there are positive, the door is unlocked. Else, the user is asked to place the hand again, to restart the process.
4. Three consecutive failures disables the door.

Suggested actions: `place hand, p1, p2, p3, unlock door, disable door`
Hint: Aim for a simpler diagram e.g., treat 'all checks pass' as one decision point, rather than each check pass/fail as a separate decision point. (estimated time: 5-7 minutes)