

### **Common Spec Rubric**

These point values and categories are taken directly from the common spec; more information about each item can be found there.

### **Code Requirements**

#### 6 pts. *Design Patterns*

The project uses six design patterns. **Each pattern must be documented** (see below).

1 pt. *Pattern is **present in design doc** (see below) (6 patterns, 1 pt. each):*

- Design pattern documented, or
- Justification of why design pattern was not appropriate

#### 5 pts. *Structure*

Points awarded based on the organizational requirements in the common spec.

#### 4 pts. *Classes*

The project contains at least 15 unique and meaningful classes. Credit will not be awarded for inner classes that are used only as listeners or classes that are not meaningful, i.e. classes that have been generated merely to satisfy this requirement. Partial points (3, 2, or 1 for 13, 10, or 5 unique and meaningful classes, respectively).

#### 1 pt. *Exceptions*

The project contains at least one user-defined exception, used appropriately.

### **Design Artifacts**

#### 4 pts. *Design Doc*

Includes each design pattern or justification. Document should be as concise as possible while still completely addressing the compliance with project requirements; the use of bullet points is highly encouraged.

3 pts. *Describes how each portion of the project spec has been implemented*

Partial credit based on the percentage of the requirements addressed

1 pt. *States known problems or deficiencies in final submission*

#### 4 pts. *UML*

Points (1 - 3) may be deducted if the UML deviates too much from final submission.

2 pts. *UML class diagram*

2 pts. *UML sequence diagram*

#### 4 pts. *Comments*

Code must include Javadoc and other code comments.

2 pts. *Comment quality* (includes Javadoc, multiline, and single-line comments)

1 pt. *Javadoc for all non-private methods*

1 pt. *Javadoc is consistent with final submission*