**Java Flooring Mastery v2.4**

| Lydia | | |
| --- | --- | --- |
| **Criteria** | **Ratings** | **Pts** |
| This criterion is linked to a Learning Outcome Specifications  All requirements in the specification are completed. | |  |  |  | | --- | --- | --- | | **20.0 pts**  **Meets Expectations** | **10.0 pts**  **Needs Improvement** | **0.0 pts**  **No Submission** | | 20.0 pts |
| This criterion is linked to a Learning Outcome Code Style  All code is written and formatted to make it easy for another developer to read, including appropriate indents, object names, and comments. | |  |  |  | | --- | --- | --- | | **10.0 pts**  **Meets Expectations** | **5.0 pts**  **Needs Improvement** | **0.0 pts**  **No Submission** | | 10.0 pts |
| This criterion is linked to a Learning Outcome Maven  The application uses Maven to manage outside libraries. | |  |  |  | | --- | --- | --- | | **5.0 pts**  **Meets Expectations** | **3.0 pts**  **Needs Improvement** | **0.0 pts**  **No Submission** | | 5.0 pts |
| This criterion is linked to a Learning Outcome DI and MVC  The application submitted works properly using dependency injection and the MVC pattern as shown in the unit. | |  |  |  | | --- | --- | --- | | **5.0 pts**  **Meets Expectations** | **3.0 pts**  **Needs Improvement** | **0.0 pts**  **No Submission** | | 5.0 pts |
| This criterion is linked to a Learning Outcome I/O Operations  The application is able to perform I/O operations to a file to store and retrieve item information. | |  |  |  | | --- | --- | --- | | **10.0 pts**  **Meets Expectations** | **6.0 pts**  **Needs Improvement** | **0.0 pts**  **No Submission** | | 10.0 pts |
| This criterion is linked to a Learning Outcome Data Types  The application uses the proper data types for all variables and correctly converts between data types when necessary. | |  |  |  | | --- | --- | --- | | **5.0 pts**  **Meets Expectations** | **3.0 pts**  **Needs Improvement** | **0.0 pts**  **No Submission** | | 5.0 pts |
| This criterion is linked to a Learning Outcome Service Layer  The application has a Service Layer that contains business logic. | |  |  |  | | --- | --- | --- | | **5.0 pts**  **Meets Expectations** | **3.0 pts**  **Needs Improvement** | **0.0 pts**  **No Submission** | | 5.0 pts |
| This criterion is linked to a Learning Outcome Lambdas and Streams  The application uses lambdas and streams in at least part of the DAO. | |  |  |  | | --- | --- | --- | | **5.0 pts**  **Meets Expectations** | **3.0 pts**  **Needs Improvement** | **0.0 pts**  **No Submission** | | 5.0 pts |
| This criterion is linked to a Learning Outcome Unit Tests  Application has full set of unit tests for DAO(s) and the Service Layer. | |  |  |  | | --- | --- | --- | | **5.0 pts**  **Meets Expectations** | **3.0 pts**  **Needs Improvement** | **0.0 pts**  **No Submission** | | 5.0 pts |
| This criterion is linked to a Learning Outcome Data Storage  Application uses a List or Map to store DAO data in memory. | |  |  |  | | --- | --- | --- | | **5.0 pts**  **Meets Expectations** | **3.0 pts**  **Needs Improvement** | **0.0 pts**  **No Submission** | | 5.0 pts |
| This criterion is linked to a Learning Outcome Input Validation  Application validates user input. | |  |  |  | | --- | --- | --- | | **5.0 pts**  **Meets Expectations** | **3.0 pts**  **Needs Improvement** | **0.0 pts**  **No Submission** | | 5.0 pts |
| This criterion is linked to a Learning Outcome Exception Handling  Application properly handles all exceptions. | |  |  |  | | --- | --- | --- | | **5.0 pts**  **Meets Expectations** | **3.0 pts**  **Needs Improvement** | **0.0 pts**  **No Submission** | | 5.0 pts |
| This criterion is linked to a Learning Outcome Basic Concepts  Apprentice can explain the foundational elements of the Java language, object-oriented design, and testing methodology. | |  |  |  | | --- | --- | --- | | **5.0 pts**  **Meets Expectations** | **3.0 pts**  **Needs Improvement** | **0.0 pts**  **No Submission** | | 3.0 pts |
| This criterion is linked to a Learning Outcome Intermediate Concepts  Apprentice can explain dependency injection, the advantages of loosely coupled code, and the MVC pattern. | |  |  |  | | --- | --- | --- | | **5.0 pts**  **Meets Expectations** | **3.0 pts**  **Needs Improvement** | **0.0 pts**  **No Submission** | | 5.0 pts |
| This criterion is linked to a Learning Outcome Application Architecture  Apprentice can explain the importance of the Service Layer in application architecture. | |  |  |  | | --- | --- | --- | | **5.0 pts**  **Meets Expectations** | **3.0 pts**  **Needs Improvement** | **0.0 pts**  **No Submission** | | 5.0 pts |
| **Total Points: 98.0** | | |

**Feedback and Areas to improve:**

Your code meets most of the expectations and you have demonstrated an excellent understanding of Java language. The test case methods are good and in line with the industry standards. The application is user-friendly and source code is consistent in most of the layers (controller, dao and service). However, the usage of access specifiers (private/public/package private) are not fully consistent.