|  |  |
| --- | --- |
| Specifications: Learner applies the specifications to the application. | Should be evident within code. |
| Dependency Injection: The application uses dependency injection. | Should be evident within code. |
| MVC: The application uses an MVC pattern. | Should be evident within code. |
| I/O Operations: The application can perform I/O operations to a file to store and retrieve data. | Should be evident within code.  Specifically, in the FileIO class and the interface DataIO that it implements. |
| Service Layer: The application has a service layer that contains business logic. | Should be evident within code.  Specifically, in the ‘Services’ class. |
| Unit Tests: The application has a full set of unit tests for the DAO and service layer. | Should be evident within code.  Note that due to the volatile nature of the file storage system and the specified nature of the class, it is not possible to perform effective unit tests on any method within the DAO layer. |
| BigDecimal: The application uses BigDecimal for all monetary calculations. | Should be evident within code.  Most notably, within the Services class. |
| Lambdas and Streams: The application uses lambdas and streams in at least part of the DAO. | Should be evident within code.  Specifically, within getInventoryReadout() in class Services, and the method offerPurchase() in class cmdUserIO. |
| Enums: The application uses enums to represent coins and their values. | Should be evident within code.  Specifically, in the Enum ‘Change’, and the methods that use it, primarily within ‘Services’. |
| Audit Log: The application includes an audit log using the Java DateTime API. | Should be evident within code.  Specifically, within the writeAuditLog method of the FileIO class. |
| Explain Service Layer: The learner can explain the role of the service layer. | A service layer allows for the separation of the core organisation and functioning of a program (the controller) and the business logic. This ensures that arbitrary or complex calculations or evaluations are kept separate from the program’s core loop, enabling more loosely coupled code and more effective encapsulation. |
| Explain Unit Testing: The learner can explain why we write unit tests. | Unit tests perform the simple function of ensuring that individual elements of the program function as intended. The value of unit tests lies in their ability to automatically recognise errors in the code’s core functionality whenever the testing suite is run. This prevents thoughtless mistakes from causing major errors to seep into the code in ways that are not detected in a timely manner. |
| Code Style: Code is written with appropriate indents, naming conventions, and comments so that other developers can read the code easily. | Should be evident within code. |