Project Report - CS1OPNU Coursework 2

Student Info

Student ID: 32803938
Module: CS1OPNU - Object-Oriented Programming
Coursework: CW2 - Project Report (Due: 5 June 2025)

Introduction

This report reflects on the development process and learning experience associated with the “Data Science Data Manager” project developed for CW1. The project involved building a modular, Java-based tool that could load, transform, analyze, and export structured data (CSV files), while incorporating three key object-oriented design patterns: Singleton, Factory, and Observer.

Design Patterns Used

1. Singleton Pattern

Class: DataManager
Purpose: Ensures a single shared instance of the dataset manager, maintaining consistent state and global configuration throughout the program.
Reflection: Implementing Singleton taught me how to control access to shared resources in a multicomponent system.

2. Factory Pattern

Class: TransformerFactory
Purpose: Dynamically creates filter operations (e.g., filter greater than, filter equals) based on input parameters.
Reflection: Using Factory helped separate object creation from logic, which made it easier to scale and test.

3. Observer Pattern

Classes: Observer, ConsoleObserver
Purpose: Enables the system to notify observers when the dataset is updated (e.g., after filtering or import).
Reflection: This pattern improved decoupling and made real-time feedback (via console updates) possible.

Challenges Faced

• Designing a reusable and modular architecture while also ensuring that it remained testable.
• Managing exceptions and validation when importing malformed CSV files.
• Implementing correct statistical logic for mean, median, and standard deviation.
• Ensuring separation of concerns among IO, core logic, and transformation components.

Learning Reflection

This project significantly enhanced my understanding of object-oriented design and how abstract design principles are implemented in real systems. Prior to this coursework, I had only basic familiarity with design patterns. Now I feel confident using them to build more maintainable and scalable applications.

I also learned how to write unit tests using JUnit, which gave me a new appreciation for the importance of testing and debugging in software development.

Use of AI Tools

I used AI (ChatGPT) in a limited and ethical way—primarily to clarify programming concepts, validate logic, and double-check my understanding of design pattern structures. I did not use AI to write the full code or report.

All submitted code is fully written and tested by myself.

Ethical and Legal Considerations

• Plagiarism Awareness: I ensured that all code was my own or adapted with clear understanding from permitted sources.
• Data Handling: I used only mock data (sample.csv) to avoid any privacy issues.
• Tool Usage: AI was used in accordance with university guidance. No prohibited auto-generation was used in the final report or source code.

Conclusion

This project offered a practical opportunity to reinforce object-oriented programming skills and software design principles. I improved my technical ability, design thinking, and problem-solving skills. The use of patterns like Singleton, Factory, and Observer greatly influenced my approach to building maintainable Java software.

References

- Oracle Java Documentation
- Gamma et al., Design Patterns: Elements of Reusable Object-Oriented Software
- JUnit 4 Official Docs
- University of Reading: Reflective Writing Guidelines