# 

# Module 4-2 Milestone Three

# Enhancement Two: Algorithms and Data Structures

Alex Mehr  
 March 25, 2025

Enhancement Two: Algorithms and Data Structures

This paper serves as an explanatory narrative accompanying the artifact enhancements for

algorithms and data structures. It justifies the artifact's inclusion in this category of my

ePortfolio and reflects on the process used to create and improve it. The discussion highlights

the learning outcomes achieved through the artifact's development.

## Artifact Selection and Purpose

## The selected artifact for this category is the Pet adoption System Program, an authentication and

authorization system designed for pet owners and pet company admins. Originally developed as part

of the IT145 Foundation in Application Development course, this Java-based application initially

contained of 3 Java files (Dog, Driver, Rescue Animal). The enhanced program contains of four

different Java programs (dog, Driver, Monkey, Rescue Animals) and development was conducted

using Eclipse IDE to write the program. The enhancements were implemented to expand the pet store

capabilities and include more pet options and information for users.

This artifact was chosen because it involves the implementation of an authentication and authorization

system, with enhancements introducing modules for monitoring animal activities, breeds, gender, age,

color and habitats. Upon login, users access role-specific data. The system design incorporates

authentication and authorization mechanisms to validate user credentials, track user interactions, and

enforce access control based on assigned roles.

The system was improved by integrating principles of input validation, default denial in security

architecture, and structured software design. These enhancements reflect a security-focused mindset

aimed at mitigating vulnerabilities, ensuring data privacy, and protecting system resources. The

program structure follows a modular approach, where source code is organized into multiple classes

and methods based on their specific functions.

## Key Improvements Include:

- Data Handling and Input Validation: The program reads external files into dynamic data structures,

such as string arrays, to process user inputs and display relevant content.

- Algorithmic Optimization: The program iterates through external data files, evaluating conditions

line by line to authenticate users and authorize actions.

- Code Structure and Readability: The source code adheres to industry best practices, including proper

naming conventions, inline documentation, and consistent formatting, making it more maintainable

and easier to read.

## Implementation of Software Engineering Principles

The artifact showcases the application of software engineering principles by organizing its

functionality into well-structured methods and classes. For instance:

- Encapsulation and Modularization: The program consists of a primary class and four

supporting classes (modules), each responsible for specific operations.

- Graphical Enhancements: To improve user experience, the system introduces a graphical component

to clear the terminal screen, display banners, and use third-party libraries for ANSI colors and text

wrapping.

- Logical Flow and Control Structures: The implementation ensures all control paths are covered, using

IF-ELSEIF and DO-TRY and CASE statements with appropriate default clauses. Loops are structured

to prevent unintended index manipulation.

## Addressing Errors and Enhancing User Experience

Error handling was a significant challenge, particularly when determining how to classify methods and

their locations in the program. To ensure the program functions correctly across different execution

environments a structured approach was adopted.

Additional improvements include:

- Platform-Specific Adjustments: A code block determines the operating system in which the program

runs, allowing it to adapt accordingly.

- User-Centric Design: Enhancements focus on usability, ensuring that screens display relevant

information based on user actions while maintaining a clean interface.

## Conclusion

The enhancements made to the Pet store System Program demonstrate the application of

algorithmic principles, structured programming, and industry best practices. The project extends

beyond simple input/output functionality, requiring in-depth research and adaptation of programming

techniques to ensure portability across different operating systems. These improvements highlight the

ability to design efficient, secure, and user-friendly software solutions aligned with professional

computing standards.