Milestone One Script for video
CS499 - Capstone

Introduction

Robert Heavner

What I Will Cover in This Code Review

- **Software Engineering and Design** – My **Rescue Animal Management System**, transitioning from Java to Python along with adding a feature to track training status.
- **Algorithms and Data Structures** – **Converting a LinkedList**-based system **into a database structure**.
- **Databases** – **Designing a fully functional SQL database** to store and analyze return data.

Milestone One Script for video
CS499 - Capstone

Category 1: Software Engineering and Design

Rescue Animal Management System (Port from Java to Python)

Existing Code (Java Version)

- The system manages rescue animals (dogs and monkeys) and tracks their training progress.

Uses four classes:

- RescueAnimal.java (Base class for all animals).
- Dog.java (Subclass for dogs, adding a breed property).
- Monkey.java (Subclass for monkeys, adding species, tail length, height, body length).
- Driver.java (Main class for user input and managing animal records).

Current Issues in the Java Code:

- No proper error handling – If a user enters invalid data, the program may crash.
- Lack of input validation – No checks for valid numbers or text.
- Repetitive logic – Some code could be simplified.
- Long code length – Java requires more lines compared to Python.
- Planned Enhancements in Python
- Error Handling & Input Validation
- Add try-except blocks to prevent crashes.
- Ensure user inputs are valid numbers or text before accepting them.
- Improve Readability
- Use better variable and function names.
- Add clear comments and docstrings for maintainability.

New Feature: Training Progress Tracking

- Add a status update function to track an animal's training progress.

Why Move to Python?

- **Shorter, cleaner code** – Python requires fewer lines to achieve the same result.
- **Easier maintenance** – Less code means fewer bugs and faster debugging.
- **Faster development** – Python allows for quick changes without recompiling.

Milestone One Script for video
CS499 - Capstone

Category 2: Algorithms and Data Structures

Converting LinkedList System into SQL Database

Existing Code (LinkedList.sln)

- The **old system** used a linked list **to manage sales data**.
- Issues with LinkedList-based system:
- Inefficient for large data – **Searching for items in a linked list is slow**.
- No built-in sorting or indexing – **Making queries difficult**.
- Not scalable – **As data grows, linked lists become harder to manage**.

Planned Enhancements: SQL Database

Move Data to SQL

- **Create database** tables to replace the linked list structure.
- Allow efficient data storage and retrieval.

Stored Procedures

- **Automate data insertion** and updating using SQL procedures.

Better Querying

- **Use SQL queries** to quickly find, sort, and filter data instead of looping through a list.

Scalability

- The **database can handle millions of records efficiently**.
- **Easy to expand with new tables and relationships**.

Milestone One Script for video
CS499 - Capstone

Category 3: Databases

Create a Fully Functional SQL Database for Return Data Analysis

Code Review (Current Return Analysis)

- In this case it is less than a review of code but more of a desired outcome.
- Let's review current goals in mind for this project by reviewing current code logic for SQL Database creation and data mining.

Current Goals:

- Design a Structured Database
- Design a real sample database with everything needs for a store to begin using as an inventory management and return system; along with having customer data and sales.
- In my case I am going to use PostgreSQL for my database, due to its powerful database engine that allows for extensions and real world integration.

Create Tables: Products, Returns, Customers, Sales.

- Establish relationships between data for better organization.

Create SQL Queries & Stored Procedures

- Stored procedures to analyze return rates by product and state.
- Functions to calculate total return percentages.

Insert Sample Data

- Populate the database with mock return records for testing.

Benefits of SQL

- Faster searches – SQL can filter data instantly.
- More accurate calculations – Reduces human errors.
- Easier reporting – Automated queries provide insights quickly.