

**PROGRAMMING PARADIGMS**

**Topic: Animal Shelter Management System**

SEM III

ACADEMIC YEAR: 2024-25

CA 3: Experiential Learning

**Group Members:**

|  |  |  |  |
| --- | --- | --- | --- |
| Sr. No. | PRN | Name of Student | Mail id |
| 1 | 23070122193 | Sanskriti Shrivastava | sanskriti.shrivastava.btech2023@sitpune.edu.in |
| 2 | 23070122195 | Saumya Kumar | saumya.kumar.btech2023@sitpune.edu.in |
| 3 | 23070122197 | Shanvi Srivastava | shanvi.srivastava.btech2023@sitpune.edu.in |
| 4 | 23070122218 | Tanisha Sirohi | tanisha.sirohi.btech2023@sitpune.edu.in |

**Problem Statement:**

**Animal Shelter Management System**

The goal of this project is to create a management system for an animal shelter that facilitates the tracking and administration of animals within the shelter. The system will allow users to add new animals, remove animals when adopted, search for animals by ID, and filter animals by species and gender. The data will be stored in a CSV file, which the program will read upon startup.

**Objectives**

* **To create a system** that allows tracking and managing various animals in a shelter.
* **Support basic operations** like adding, removing, filtering, and displaying animal information.
* **Store and retrieve data** from a CSV file, ensuring persistence of information.

**Brief Explanation:**

This C++ program simulates an animal shelter management system where data about animals is read from a CSV file and stored as objects. The system allows for adding, removing, and viewing animals, as well as filtering based on specific criteria such as species or adoption month. Animals can also be categorized by species, and information regarding whether an animal has been returned is tracked.

**How We Built It**

We used two important programming concepts: **inheritance** and **polymorphism**. These concepts allow us to create a flexible and efficient framework that can easily adapt to the needs of different animal species.

**Key Features**

1. **Class Inheritance Structure**:

* **Base Class (Animal)**: Represents general animal information, with fields like ID, intake details, shelter info, and movement/adoption details.
* **Derived Classes**: Specialized for different animals like Dog, Cat, Rabbit, Hamster, and Tortoise. These inherit from the base Animal class.

1. **Polymorphism**:

* The program utilizes polymorphism to allow derived classes (specific animal types) to override base class methods. This enables dynamic method resolution, allowing the program to treat different animal types uniformly while still invoking their specific behaviours.

1. **File Handling**:

* Reads and writes data to a CSV file. The readAnimalsFromCSV() function loads existing animals into the program, while new animals can be added and saved back to the file.

1. **Data Filtering and Search**:

* **Find by ID**: Allows searching for animals by their unique ID.
* **Filter by Species**: Filters animals based on their species (Dog, Cat, Rabbit, Hamster, or Tortoise).
* **Count by Adoption Month**: Tracks how many animals were adopted in a specific month.
* **Return Status Check**: Verifies whether an animal has been returned to the shelter and provides the return details.

1. **Dynamic Memory Management**:

* Animals are stored as pointers in a vector, and dynamic memory allocation is used. After program execution, memory cleanup is performed to avoid leaks.

1. **Menu-Driven Interface**:

* A user-friendly console interface allows the user to choose different operations like displaying all animals, adding new ones, or removing animals based on their ID.

1. **Counting and Analysis**:

* The system can count animals by species and display this information.
* Animals can be added and removed from the system, and the updated information is stored back in the CSV file.

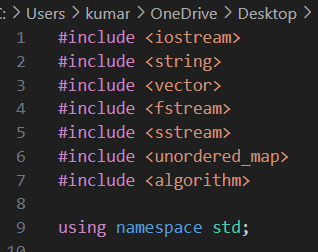
**Conclusion**

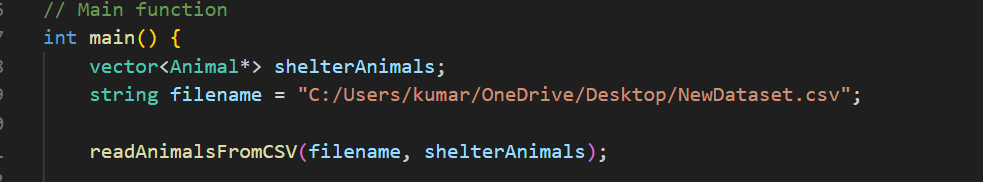
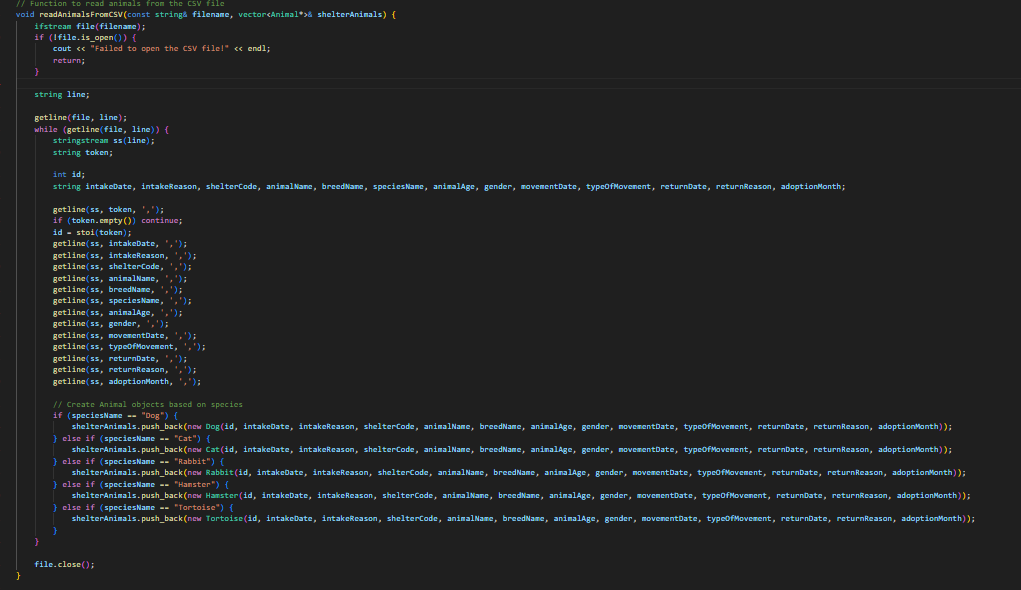
The C++ program for the animal shelter management system demonstrates effective use of object-oriented programming principles, particularly inheritance and polymorphism, to enhance code reusability and maintainability. Its features, including file handling, data filtering, and a user-friendly menu, facilitate efficient management of animal records.

By employing dynamic memory management, the program optimizes resource usage. Overall, this system streamlines operations in animal shelters and supports animal welfare, with opportunities for further enhancements to strengthen its functionality.

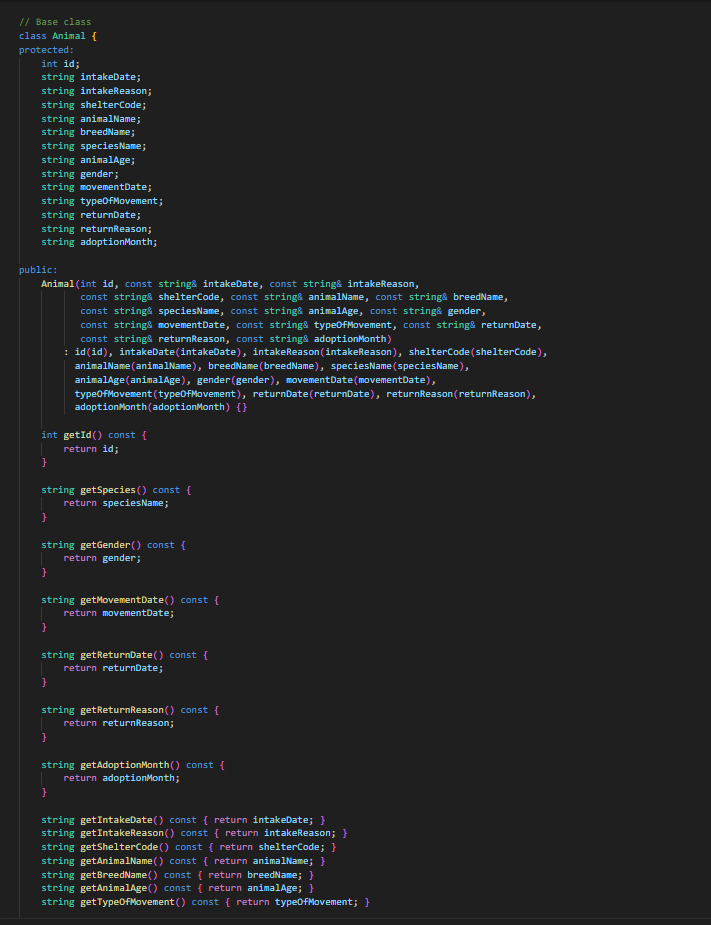
**Code snippets:**

**Code snippet for database management-**



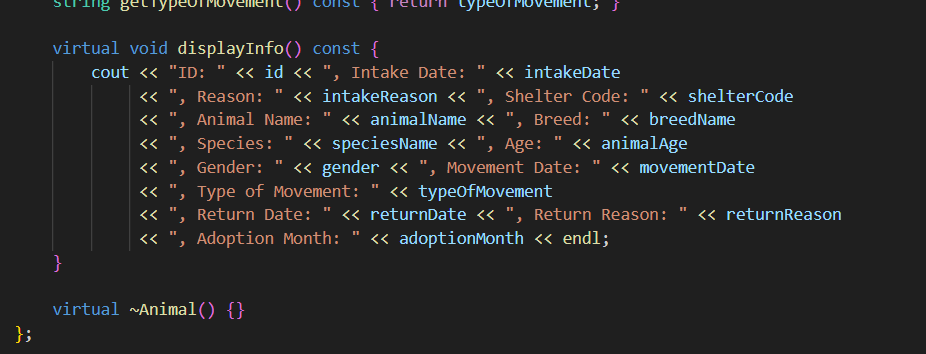


**Code snippet for inheritance-**

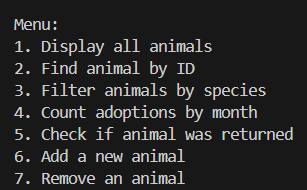




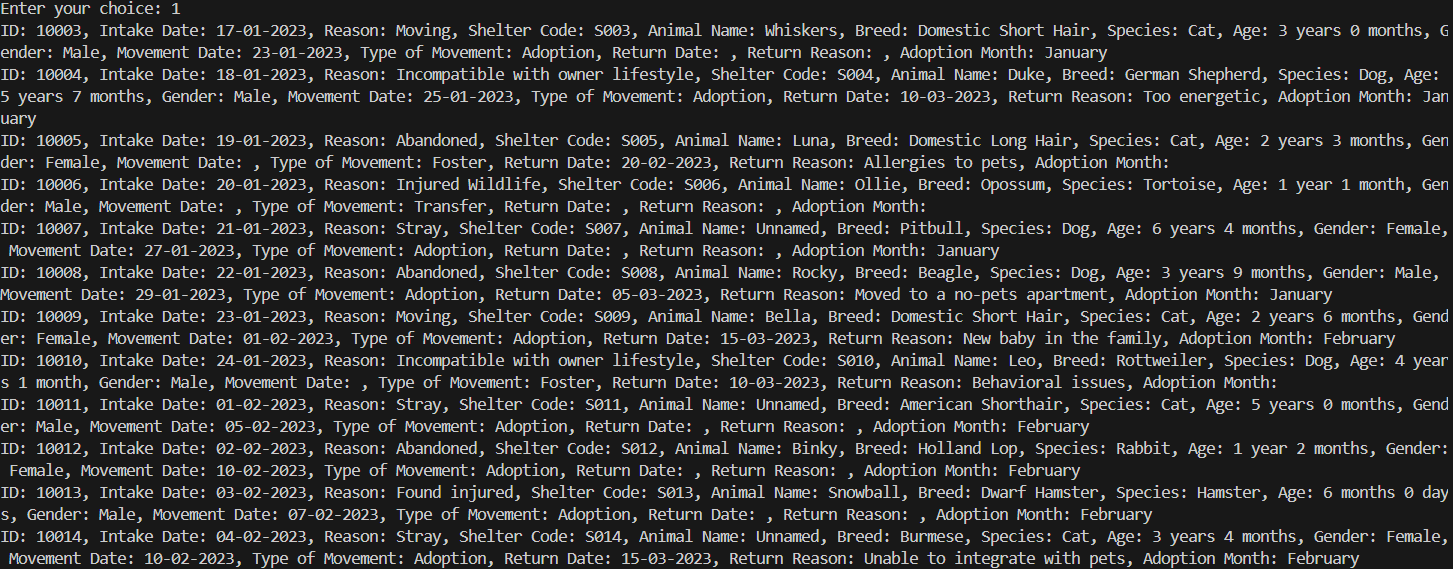
**Code snippet for polymorphism**



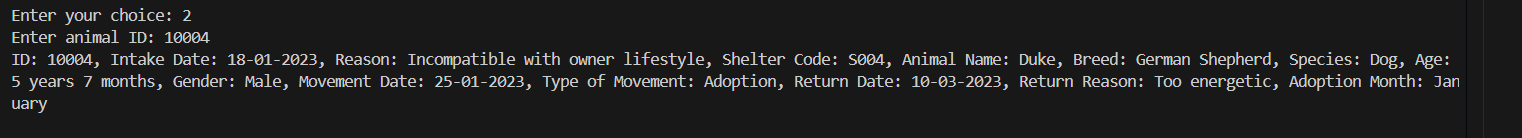
**Input/Output:**



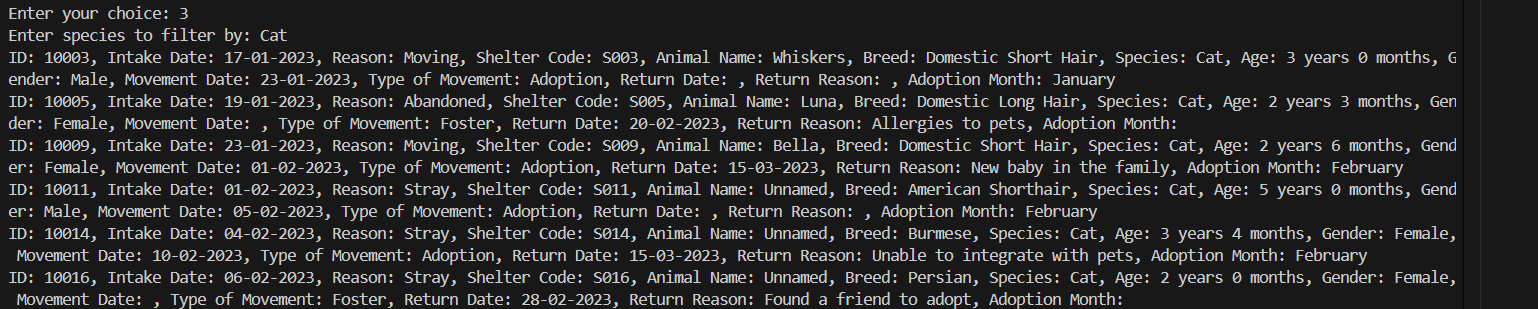
**CHOICE 1** – Entire dataset is displayed.



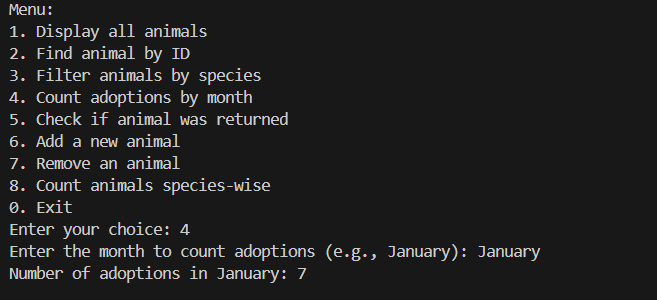
**CHOICE 2**- Finds animals by ID to get all the details of the animal

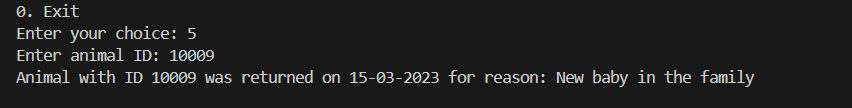


**CHOICE 3** – If the visitor wants to adopt an animal of a particular species, this feature displays all the animals available in the said species.

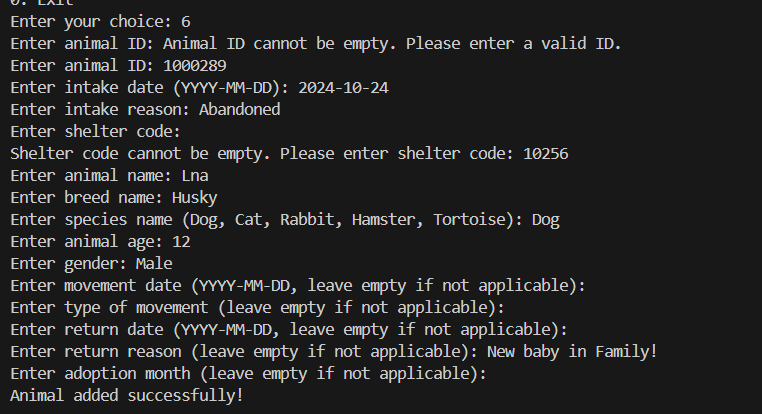


**CHOICE 4** – Checks how many adoptions were made in a particular month.

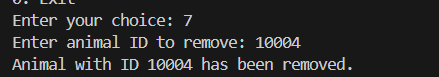


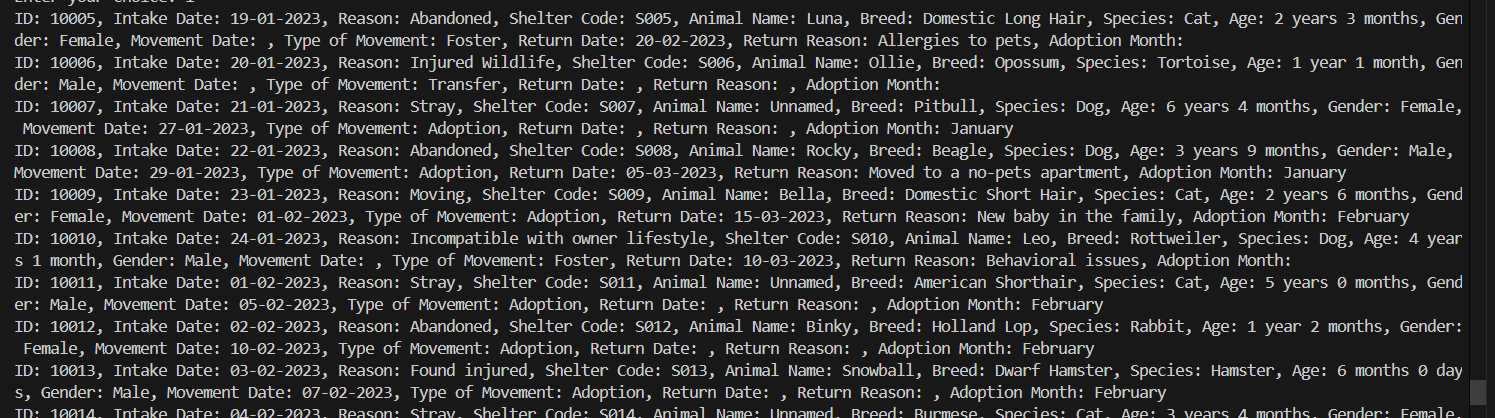
**CHOICE 5-** Check if the animal was returned to the shelter and if it was indeed returned what was the reason

**CHOICE 6-** Adds a new animal by prompting for details. Some fields are mandatory, and if the optional fields are left empty, "N/A" will be automatically filled in.

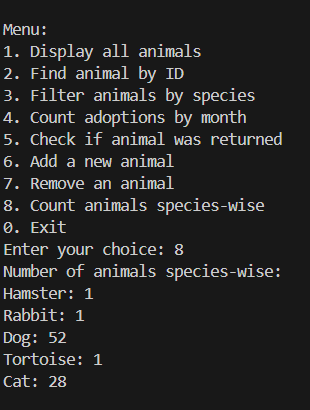


**CHOICE 7-** If the animal is adopted, its details are removed from the dataset



10004 ID animal has been removed.

**CHOICE 8-** Counts the number of animals available in the shelter specie wise.



**Github repository link:**

https://github.com/saumya08-07/Animal-Shelter-Management-System