**DESIGN DOCUMENT ASSIGNMENT-5**

**AREEN PATIL – IMT2023013**

**1. Introduction**

The code -assignment-5 is designed to manage and track species in a zoo, including mammals, reptiles, birds, and aquatic species. It allows adding, showing, and updating species, including tracking their diet type or feeding preferences.

**2. Objective**

My code allows the zoo administration to:

* Add species to the zoo, categorizing them as mammals, reptiles, birds, or aquatic species.
* Track their population and feeding characteristics.
* Ensure no conflicting species data (e.g., same species name but with different feeding preferences) can be added.

**3. Class Design**

* 1. **Base Class: Zoo\_species**
  + Stores species name, species count, and category name and Provides basic operations like setting the name and count of species.
* **Key Attributes**:
  + string Species\_name: Name of the species.
  + int Species\_count: Number of individuals in the species.
  + char Category\_name: Category type (e.g., 'M' for mammal, 'R' for reptile).
* **Key Functions**:
  + Set\_name(string name): Set the name of the species.
  + Set\_count(int count): Set the count of the species.
  + increment\_coutner(int count): Increments the species count.

**3.2 Derived Classes**

**3.2.1 Mammal Class**

* + Manages mammal species in the zoo.
* **Key Attributes**:
  + enum diet\_type Diet\_type: Defines whether the mammal is herbivore or carnivore.
* **Key Functions**:
  + Set\_diet\_type(enum diet\_type): Sets the mammal's diet type.
  + get\_diet\_type(): Retrieves the diet type as a string.
  + Print(): Prints species information, including diet type, to the console.
  + Print\_to\_file(ofstream&): Prints species information, including diet type, to a file.

**3.2.2 Reptile Class**

* + Manages reptile species in the zoo.
* **Key Attributes**:
  + enum size\_type Feed\_size: Defines whether the reptile's feed size is small, medium, or large.
* **Key Functions**:
  + Set\_feed\_size(enum size\_type): Sets the reptile's feed size.
  + get\_feed\_size(): Retrieves the feed size as a string.
  + Print(): Prints species information, including feed size, to the console.
  + Print\_to\_file(ofstream&): Prints species information, including feed size, to a file.

**3.2.3 Bird Class**

* + Manages bird species in the zoo.
* **Key Attributes**:
  + enum bird\_feed\_type Bird\_feed: Defines the bird's feeding type (grain, insect, or fish).
* **Key Functions**:
  + Set\_bird\_feed(enum bird\_feed\_type): Sets the bird's feed type.
  + get\_bird\_feed(): Retrieves the bird feed type as a string.
  + Print(): Prints species information, including bird feed type, to the console.
  + Print\_to\_file(ofstream&): Prints species information, including bird feed type, to a file.

**3.2.4 Aquatic Class**

* + Manages aquatic species in the zoo.
* **Key Attributes**:
  + enum bird\_feed\_type Bird\_feed: Defines the bird's feeding type (fishfood, livefish or plants).
* **Key Functions**:
  + Set\_bird\_feed(enum bird\_feed\_type): Sets the bird's feed type.
  + get\_bird\_feed(): Retrieves the bird feed type as a string.
  + Print(): Prints species information, including bird feed type, to the console.
  + Print\_to\_file(ofstream&): Prints species information, including bird feed type, to a file.

**4. System Functions**

**4.1 add Function:**

**Use:**

* + Accepts the category character (M for mammals, B for birds, R for reptiles, Q for aquatic), species name, count, and diet/feed type as arguments.
  + Checks whether the species already exists in the vector v of Zoo\_species\* based on name and diet/feed type.
  + If the species exists with the same characteristics, the count is incremented.
  + If the species exists with conflicting characteristics, an error is printed.
  + If the species does not exist, a new object is created, populated with the given details, and added to the vector v.

**4.2 show Function:**

**Use:**

1 . Accepts a category character and prints the corresponding class’s details.

2 . Iterates through the vector v and prints species details (name, count, feed/diet type) for the specified category

3. If the category character is A, it prints all species.

* 1. **del Function :**

**Use**:

1. If the specified species is found and the count is greater than the number to be deleted, the count is reduced. If the count becomes 0, the species is removed from the vector.

2. Error handling for invalid category, species not found, or insufficient count.

**5. Key Enumerations**

**5.1 enum diet\_type**

* Used in Mammal class to differentiate between herbivore and carnivore.

**5.2 enum size\_type**

* Used in Reptile class to categorize feed size as small, medium, or large.

**5.3 enum bird\_feed\_type**

* Used in Bird class to categorize bird feed as grain, insect, or fish.

**5.4 enum aqua\_feed\_type**

* Used in Aquatic class to categorize aquatic feed as fishfood, livefish, or plants.

**6. Main Loop**

* Continuously prompts for user commands (add, show, delete).
* When the add command is entered, it adds a species to the vector.
* When the show command is entered, it displays species based on the category.
* When the delete command is entered, it deletes or decreases the count of a species.
* The loop ends when an unrecognized command is given, triggering the data to be written back to the file and the program to terminate.

7. **Structure of the Assignment**

I have created a Classes.h that stores declaration for all the classes and the enum . Then I have separate files for the implementation for each of the classes namely, Zoo\_species.cpp, Aquatic.cpp, Bird.cpp, Mammal.cpp and Reptile.cpp. These are all used included and used in the main.cpp which has the functions add, show and del and the driver code for taking the commands and making the function calls.

8. **Outputs**

added successfully -- Indicating the species was added successfully.

deleted successfully – Indicating the species was deleted successfully.

Error – indicating failure in the execution of the command, followed by the exact reason for the failure.