1. In this assignment you will create a pet inventory application by fulfilling the requirements given below. You must first create the classes described below. Then write a driver program that will use these classes to build an application for a Pet Store. All pet data is stored in a file called petdata.dat.
2. Each time you run the program, the user is allowed to repeatedly select from the four choices described below:
3. Add a new pet
4. Compute and print current weight together with the age (in days)of all pets in the file
5. Print all pet information for every pet in the file
6. Exit the application
7. The first time you execute the program, your file will have no pets in it. Second time onwards, all pet data from previous executions of the program, should be available

Pet Application

* The base class Pet has the following attributes:
  + pet’s name (String)
  + date of birth (date),
  + birth weight (float), and
  + owner (Person).
* The child classes Mammal, Fish and Amphibian that inherit from Pet and have the following additional properties
* Mammals - litter size (int), hasClaws (boolean)
* Fish - scale condition (String), length (float)
* Amphibian - isVenomous (boolean).
* All these classes have the \_\_init\_\_ method as well as the **get** and **set** methods. In addition, they have additional methods required to complete the application as described below.
* Current weight is calculated as follows:
* **Mammal:** Weight increases by 8% every 50 days for the first 300 days. Weight is constant after that. Assume that the increase occurs on the 50th day only.
* **Fish:** Weight increases by 4% every 90 days for the first 360 days and then stays constant. Assume that the increase occurs on the 90th day only.
* **Amphibians:** Weight increases by 5% every 120 days for the first 360 days and then increases by 3% every 120 days for the next 240 days and then stays constant. Assume that the increase occurs on the 120th day only.
* The Person class is as described below:

It contains two instance variables: one each for the name, address of the owner. It has a **\_\_init\_\_** method as well as **get** and **set** methods for all the instance variables. You may use aggregation or composition to include a Person object to your Pet data.

* The driver program does the following:

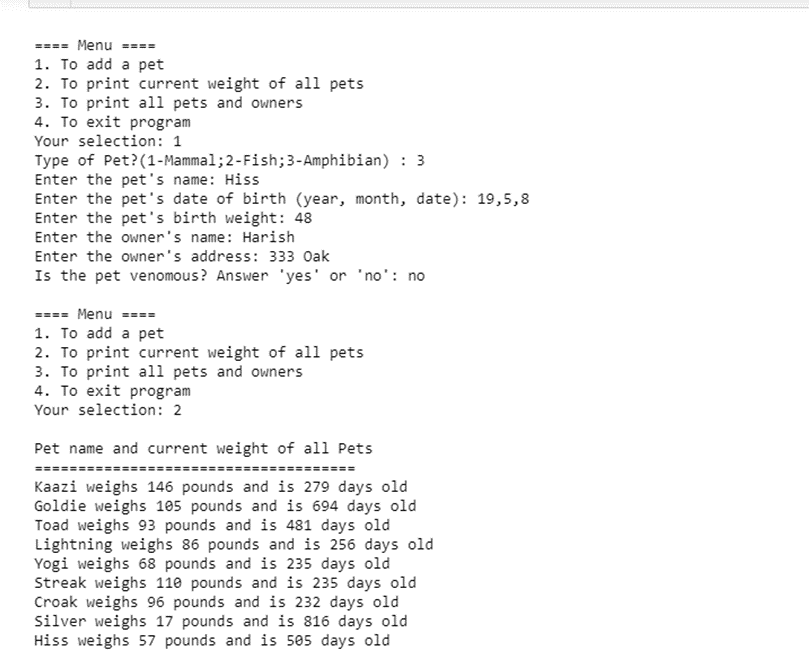
1. Accepts input for new pets
2. Lists the current weight of all the pets together with pet name. (The weight is rounded down to the nearest integer).
3. Displays all pet information for all the pets in the system. This includes owner details as well as data specific to pet type.
4. Exits the system after writing all information to a file.

Any pet data from earlier executions of the application must be available in subsequent executions.

* Additional Requirements:
  + Your program should use exception handling to validate user input for date of birth and weight. If the user enters a non-numeric value for either, or if the month or day for the date of birth is an invalid value, print a message and exit the program.
  + Your program should also ensure that the user enters valid values for the menu options as well as the choice of pet type. If the user enters incorrect options for either the menu option or the pet type, your program should display an appropriate message and display the menu or the choice of pet types again.
* Using dates in Python.
  + To complete this assignment, you will need the date class from the datetime module.
  + The user should enter year, month and date (in that order) separated by commas. If the year is not 2 characters long, raise a ValueError. Otherwise prefix ‘20’ to the year. This can then be converted to a 4-digit numeric value later.
  + Ensure that both day and month are 2-digits by padding with zeroes if necessary
  + Compute the age (in days) by subtracting the dob from today’s date.
  + Compute the current weight as described earlier.

**Submit a single notebook containing all class definitions together with the driver code. You do not need to zip the file. Name the file with our name and the chars ’hw4’.**

**The screen shots below show how your program should function with correct input. Screenshots for handling incorrect input are not shown. However, as described earlier, your code should use exception handling to validate user input.**

****

