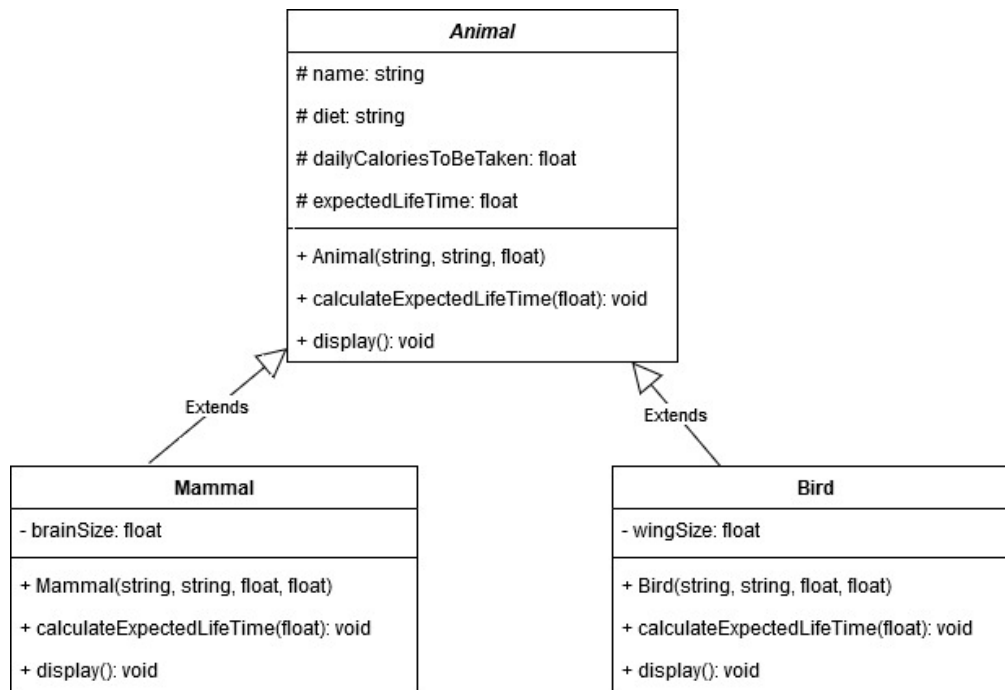


**ATILIM UNIVERSITY**  
**CMPE 225 – OBJECT-ORIENTED PROGRAMMING**  
**HOMEWORK II**

**Instructors:** Çiğdem TURHAN, Güler KALEM

**Assistants:** Buğra Yener ŞAHİNOĞLU, Gizem ALTINAY

**Due Date:** 10.05.2020 23:59



Your company wants you to develop an animal information storage system. System only stores birds and mammals in different classes and those classes are inherited from an abstract class called **Animal** as seen in class diagram.

**Animal** class is an abstract class. It has 2 string data members called name and diet, and 2 float data members called dailyCaloriesToBeTaken and expectedLifeTime. Its constructor takes three parameters and assign them to name, diet and dailyCaloriesToBeTaken data members. In the constructor, expectedLifeTime's value is assigned as 0. It has also 2 pure virtual member functions.

**Mammal** class has a float data member called brainSize. Its constructor takes four parameters and assigns them to the related data members. Override the display function in the Mammal class to display the properties of the animal such as:

Name: *Animal name*

Diet: *Diet*

Brain Size: *X m3*

Expected Lifetime: *X.XX year*

Also override the calculateExpectedLifeTime function in the Mammal class. In this function calculate the expected lifetime of the animal with following formula and assign it to expectedLifeTime.

$$ELT = \frac{param * 20.0}{dailyCaloriesToBeTaken}$$

**Bird** class has a float data member called wingSize. Its constructor takes four parameters and assigns them to the related data members. Override the display function in Bird class to display the properties of the animal such as:

Name: *Animal name*

Diet: *Diet*

Wing Size: *X m*

Expected Lifetime: *X.XX year*

Also override the calculateExpectedLifeTime function in the Bird class. In this function calculate the expected lifetime of the animal with the following formula and assign it to expectedLifeTime.

$$ELT = \frac{param * 10.0}{dailyCaloriesToBeTaken}$$

In the main function create an array of at most 10 elements of that can store Mammal or Bird type objects to enable polymorphism. In a loop, ask to user "What do you want to do?" If user enters 1, add a new animal to the array, if 2 is entered, remove the last animal from the array, if 3 is entered, display all of the information about the animals in the array.

For adding an animal ask to user "Which species do you want to add?" If user enters 1, input the information for a Mammal object and create it, if user enters 2, get the information for a Bird object, and create it. For both cases, calculate the lifetime of the animals by calling the calculateExpectedLifeTime function. Remember to put all the created animals into the array.

For removing an animal, delete the last item from the array and display its properties by calling the display function.

For display all animals, display all items in the array by calling the display function.

To end the loop, ask to user "Enter e for exit:" If user enters letter 'e', end the loop, otherwise loop continues.

Remember to delete all the newly created objects before the program ends.

### Sample Run:

```
What do you want to do?(1 for add an animal, 2 for remove last added animal, 3 display all):1
Which species do you want to add?(1 for mammals, 2 for birds):1
Enter its name:Cat
Enter its diet:Carnivore
Enter its brain size:0.08
Enter the daily calories it should take:2000
Enter the daily calories it takes:1900
Enter e for exit:c
What do you want to do?(1 for add an animal, 2 for remove last added animal, 3 display all):1
Which species do you want to add?(1 for mammals, 2 for birds):2
Enter its name:Hummingbird
Enter its diet:Herbivore
Enter its wing size:0.10
Enter the daily calories it should take:3000
Enter the daily calories it takes:2500
Enter e for exit:c
What do you want to do?(1 for add an animal, 2 for remove last added animal, 3 display all):3
All animal in stack:
1. animal:
Name:Cat
Diet:Carnivore
Brain Size: 0.08 m3
Expected Life Time:19.00 year

2. animal:
Name:Hummingbird
Diet:Herbivore
Wing Size: 0.10 m
Expected Life Time:8.33 year

Enter e for exit:c
What do you want to do?(1 for add an animal, 2 for remove last added animal, 3 display all):2
Removed Animal:
Name:Hummingbird
Diet:Herbivore
Wing Size: 0.10 m
Expected Life Time:8.33 year
Enter e for exit:c
What do you want to do?(1 for add an animal, 2 for remove last added animal, 3 display all):3
All animal in stack:
1. animal:
Name:Cat
Diet:Carnivore
Brain Size: 0.08 m3
Expected Life Time:19.00 year

Enter e for exit:e

-----
Process exited after 451 seconds with return value 0
```

### IMPORTANT NOTES:

- Cheating will NOT be tolerated. Special software will be used to verify if the submitted homework is your original work or copied from someone/somewhere else. If any case of cheating is detected, at any time, **you will get ZERO from your homework.**
- You should upload your homework file to the Moodle system.
- The name of your homework file should be **"StudentID\_HwNumber.cpp"**, for example: "11122233\_hw1.cpp"
- You should use indentation and comments in your code.
- Late submissions will be evaluated by using formula  $100-20*d^2$  where d is the number of late submission days.

- If there is any type of compiling error or runtime error seen in your code, your code evaluated by the half of the maximum evaluation score. (Late submissions will change the maximum evaluation score.)