COSC 1320, Programming Assignment 2

(Deadline: Tuesday, June 14, 2016, 11:59PM)

For this assignment you will be tasked with creating several classes. 3 abstract classes and 3 concrete classes.

The base class will be class `Animal`. This will be an abstract class following are its required specifications

- `public static` variable `count` that counts number of animals created
- default constructor that adds one to the `count`.
- pure `virtual` function `speak()` that returns a string.

There will be an class `Canine`. It inherits from `Animal`.

- `public static` variable, `count`, that counts number of `Canine`s created.
- default constructor that adds one to the `count`.
- implement the `speak()` to return string “`WOOF`”.

There will be an class `Feline`. It also inherits from `Animal`.

- `public static` variable, `count`, that counts number of `Feline`s created
- default constructor that adds one to the `count`.
- implement the `speak()` to return string “`PURR`”.

There will be a class `Dog`. It inherits from `Canine`.

- `private string` variable called `name`.
- default constructor that sets the `name` variable to “`dog`”.
- constructor that takes a single `string` parameter and sets that parameter as the `name`.
- `getName` function that returns the `name` of the `Dog`.

There will be a class `Wolf`. It inherits from `Canine`.

- default constructor that is empty.
- `howl` function that returns a string “`HOWL`”.

There will be a class `Cat`. It inherits from `Feline`.

- `private string` variable called `name`.
- default constructor that sets the `name` variable to “`cat`”.
- constructor that takes a single `string` parameter and sets the parameter as the `name`.
- `getName` function that returns the `name` of the `Cat`.

Some things to note:

- Note there is no setName function. This is intentional. The logic is that a Dog/Cat's name is set only once and will never be changed.
- Please note that base class default constructors are called before the beginning of a derived class's constructor. This means you only need to handle static counts for their respective classes (so you do not have to worry about Animal count from the Canine class).

Please also create a main.cpp file that will test the following:

- Create a Dog, Wolf and Cat, using only default constructors.
- Create a Dog and Cat with a name.
- Create an Animal * array and instantiate and store at least a Dog, a Cat and a Wolf.
- Use for loop to call speak on each of the animals stored in Animal array.
- Call howl function.
- Show all counts for Animal, Canine and Feline

Grading Criteria

1. You must submit your assignment through Blackboard Learn assignment system. Submissions via message are not going to be accepted.
2. You must add comment lines (**//explain the statements briefly**) in your program code. If you do not add comment lines, TAs are going to **deduct 10 points**.
3. If you do not use proper indentation, TAs are going to **deduct 10 points**.
4. If your program **cannot be compiled**, TAs are going to **deduct at least 40 points**.
5. If your program **terminates at run time**, TAs are going to **deduct at least 40 points**.
6. You must turn all relevant .h and .cpp files including the main.cpp file.
7. When you are ready to turn in all your files, please put them **IN ONE FILE** called **PeopleSoftIDFirstNameLastNamePA2.zip**. Please submit .zip file only. Zip file will be the only method of accepted submission.
8. If your file cannot be opened or your file is corrupted, your grade will be 0 (zero).
9. **Abstract Class Animal (12 points)**

    a. static variable count 4 **points.**

    b. default constructor **4 points**

    c. pure virtual function **4 points**

10. **Class Canine (12 points)**

    a. static variable count **4 points.**

    b. default constructor **4 points.**

    c. function implemented **4 points.**

11. **Class Feline (12 points)**

    a. static variable count **4 points.**

    b. default constructor **4 points.**

    c. function implemented **4 points.**

12. **Class Dog (16 points)**

    a. private variable name **4 points.**

    b. default constructor **4 points.**

    c. constructor with string parameter **4 points.**

    d. getName function **4 points.**

13. **Class Wolf (8 points)**

    a. default constructor **4 points.**

    b. howl function **4 points.**

14. **Class Cat (16 points)**

    a. private variable name **4 points.**

    b. default constructor **4 points.**

    c. constructor with string parameter **4 points.**

    d. getName function **4 points.**

15. **Main.cpp (24 points)**
    a. use all default constructors **4 points.**
    b. use all name constructors **4 points.**
    c. create Animal * array **4 points.**
    d. for loop for speak **4 points.**
    e. use howl function **4 points.**
    f. show all static counts **4 points.**
16. Multiple submissions will be allowed until the due date. Your final grade will always be based on the last submission.
17. Deductions for the late submission: (One day, 20%), (Two days, 30%), (Three days, 40%), and so forth…
18. **Any kind of cheating or plagiarism will result in at least a 0 (zero) GRADE**.

☺ Good Luck and Have Fun ☺