

```
1 //=====
2 // Name      : Animal.h
3 // Author    : Sotheanith Sok
4 // Version   : 1.0
5 // Description : This is an abstract class contains: virtual destructor, count of ↗
6 //             all object created from Animal, operator<< overload,...etc.
7 //=====
8 #ifndef ANIMAL_H_
9 #define ANIMAL_H_
10 #include <string>
11 class Animal
12 {
13 public:
14     virtual std::string talk() = 0;
15     //Precondition:
16     // _None.
17     //Postcondition:
18     // _Virtual method used to initilize how the Animal's talk.
19     virtual std::string move() = 0;
20     //Precondition:
21     // _None.
22     //Postcondition:
23     // _Virtual method used to initilize how the Animal's move.
24     virtual ~Animal() = 0;
25     //Precondition:
26     // _None.
27     //Postcondition:
28     // _Virtual destructor used to deallocated the memory used to initilize ↗
29     // variable.
30
31     std::string getAnimalType();
32     //Precondition:
33     // _None.
34     //Postcondition:
35     // _Return the name of this animal.
36
37     int getCount();
38     //Precondition:
39     // _None.
40     //Postcondition:
41     // _Return the number of this object existed.
42
43     friend std::ostream& operator<<(std::ostream& os, Animal& obj);
44     //Precondition:
45     // _None.
46     //Postcondition:
47     // _Return os contains animalType, animal's talk, animal's move.
48 protected:
49     std::string *animalType = NULL;
50     //Precondition:
```

```
51     // _None.
52     //Postcondition:
53     // _This pointer pointed to a animalType string.
54
55     static int count;
56     //Precondition:
57     // _None.
58     //Postcondition:
59     // _This variable created on the number of Animal existed.
60
61 };
62 #endif
63
64
```