```cpp
//============================================================================
// Name        : Animal.cpp
// Author      : Sotheanith Sok
// Version     : 1.0
// Description : This is an abstract class contains: virtual destructor, count of all object created from Animal, operator<< overload,...etc.
//============================================================================

#include "Animal.h"
#include<iostream>

//Precondition:
//  _None.
//Postcondition:
//  _This varaible created on the number of Animal existed.
int Animal::count = 0;

//Precondition:
//  _None.
//Postcondition:
//  _Virtual destructor used to deallocated the memory used to initilize variable.
Animal::~Animal()
{
    delete animalType;
    count--;
}

//Precondition:
//  _None.
//Postcondition:
//  _Return the name of this animal.
std::string Animal::getAnimalType()
{
    return "["+*animalType+"]";
}

//Precondition:
//  _None.
//Postcondition:
//  _Return the number of this object existed.
int Animal::getCount()
{
    return count;
}

//Precondition:
//  _None.
//Postcondition:
//  _Return os contains animalType, animal's talk, animal's move.
std::ostream & operator<<(std::ostream & os,  Animal & obj)
{
    os << *(obj.animalType) << ", " << obj.talk() << ", " << obj.move();
```

```
52      return os;
53  }
54
```