

Lesson 3:
Advanced OOP

SEARCH

RESOURCES

CONCEPTS

1. Polymorphism and inheritance

2. Bjarne on Inheritance

3. Inheritance

4. Access Specifiers

5. Exercise: Animal Class

6. Composition

7. Exercise: Class Hierarchy

8. Exercise: Friends

9. Polymorphism: Overloading

10. Polymorphism: Operator Overlo...

11. Virtual Functions

12. Polymorphism: Overriding

13. Override

14. Multiple Inheritance

15. Generic Programming

16. Bjarne on Generic Programming

17. Templates

18. Bjarne on Templates

19. Exercise: Comparison Operation

20. Deduction

21. Exercise: Class Template

22. Summary

23. Bjarne on Best Practices with Cla...

Multiple Inheritance

SEND FEEDBACK

Exercise

In the code below, the `Dog` class inherits from both `Animal` and `Pet`. Once you have examined the code below, add a `Cat` class that also inherits from `Animal` and `Pet`.

The `Cat` class should have the attribute `string color`. Design the `Cat` class to pass the tests in the `main()` function.

In []:

```
#include <iostream>
#include <string>
#include <assert.h>

class Animal {
public:
    double age;
};

class Pet {
public:
    std::string name;
};

// Dog derives from *both* Animal and Pet
class Dog : public Animal, public Pet {
public:
    std::string breed;
};

class Cat : public Animal, public Pet{
public:
    std::string color;
};

int main()
{
    Animal animal;
    Pet pet;
    Dog dog;
    Cat cat;
    cat.color == "black";
    cat.age == 10;
    cat.name == "Max";
    assert(cat.color == "black");
    assert(cat.age == 10);
    assert(cat.name == "Max");
}
```

Compile & Execute

Explain

Loading terminal (id_ey2gcrt0), please wait...

↑ Menu

Shrink

NEXT