

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...

Exercise: Friends

https://youtu.be/GxdPV4mz7wg

Fri

end

In C++, `friend` classes provide an alternative inheritance mechanism to derived classes. The main difference between classical inheritance and friend inheritance is that a `friend` class can access private members of the base class, which isn't the case for classical inheritance. In classical inheritance, a derived class can only access public and protected members of the base class.

Instructions

In this exercise you will experiment with friend classes. In the notebook below, implement the following steps:

1. Declare a class `Rectangle`.

2. Define a class `Square`.

3. Add class `Rectangle` as a friend of the class `Square`.

4. Add a private attribute `side` to class `Square`.

5. Create a public constructor in class `Square` that initializes the `side` attribute.

6. Add private members `width` and `height` to class `Rectangle`.

7. Add a `Rectangle()` constructor that takes a `Square` as an argument.

8. Add an `Area()` function to `class Rectangle`.

Saving Graffiti Recording

Please wait...

Menu

Expand

NEXT