

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

SEND FEEDBACK

In 1 | 1:

```
// Example solution for Rectangle and Square friend classes
#include <assert.h>

// Declare class Rectangle
class Rectangle;

// Define class Square as friend of Rectangle
class Square {
// Add public constructor to Square. Initialize side
public:
    Square(int a) : side(a) {}

private:
// Add friend class Rectangle
    friend class Rectangle;
// Add private attribute side to Square
    int side;
};

// Define class Rectangle
class Rectangle {
// Add public functions to Rectangle: area() and convert()
public:
    Rectangle(const Square& a):
        int Area() const;

private:
// Add private attributes width, height
    int width {0};
    int height {0};
};

// Define a Rectangle constructor that takes a Square
Rectangle::Rectangle(const Square& a) : width(a.side), height(a.side) {}

// Define Area() to compute area of Rectangle
int Rectangle::Area() const
{
    return width * height;
}

// Update main() to pass the tests
int main()
{
    Square square(4);
    Rectangle rectangle(square);
    assert((rectangle.Area() == 16));
}
```

Compile & Execute

Explain

Loading terminal (id_jyjd706), please wait...

Show Solution

Loading [MathJax]/extensions/Safe.js

Menu

Shrink

NEXT