

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...

Composition

SEND FEEDBACK

In 1 |>

```
// Example solution for Circle class
#include <iostream>
#include <math>
#include <cassert>
// Define PI
#define PI 3.14159;

// Define LineSegment struct
struct LineSegment {
    // Define protected attribute length
public:
    double length;
};

// Define Circle class
class Circle {
public:
    Circle(LineSegment& radius):
        double Area();

private:
    LineSegment& radius_;
};

// Declare Circle class
Circle::Circle(LineSegment& radius) : radius_(radius) {}

double Circle::Area()
{
    return pow(Circle::radius_.length, 2) * PI;
}

// Test in main()
int main()
{
    LineSegment radius {3};
    Circle circle(radius);
    assert(int(circle.Area()) == 28);
}
```

Run Code

Explain

Loading terminal (id_39egm48), please wait...

Show Solution

Loading [MathJax]/extensions/Safe.js

Menu

Shrink

NEXT