

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...

Virtual Functions

SEND FEEDBACK

In 1 | 1: ▶

// Example solution for Shape Inheritance
#include <assert.h>
#include <cmath>

// TODO: Define pi
class pi{
 // TODO: Define the abstract class Shape
 // TODO: Define public virtual functions Area() and Perimeter()
 // TODO: Append the declarations with = 0 to specify pure virtual functions
};

// TODO: Define Rectangle to inherit publicly from Shape
class Rectangle{
 // TODO: Declare public constructor
 Rectangle(int width, int height) : width(width), height(height){}
 // TODO: Override virtual base class functions Area() and Perimeter()
 public:
 virtual int Area(){return width*height;}
 virtual int Perimeter(){return (width+height)*2;}
 // TODO: Declare private attributes width and height
 private:
 int width, height;
 int Area;
};

// TODO: Define Circle to inherit from Shape
class Circle{
 // TODO: Declare public constructor
 Circle(int radius) : radius(radius){}

 // TODO: Override virtual base class functions Area() and Perimeter()
 public:
 virtual float Area(){return radius*radius*3.141592;}
 virtual float Perimeter(){return 2*3.141592*radius;}
 // TODO: Declare private member variable radius
 private:
 int radius;
};

// Test in main()
int main() {
 double epsilon = 0.1; // useful for floating point equality

 // Test circle
 Circle circle(12.31);
 assert(abs(circle.Perimeter() - 77.35) < epsilon);
 assert(abs(circle.Area() - 476.06) < epsilon);

 // Test rectangle
 Rectangle rectangle(10, 6);
 assert(rectangle.Perimeter() == 32);
 assert(rectangle.Area() == 60);
}

Compile & ExecuteShow SolutionExplain

Loading terminal (id_oxz5p8d), please wait...

Loading [MathJax]/extensions/Safe.js

↑ Menu🔍 ShrinkNEXT