

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...

Override

SEND FEEDBACK

In 1 1: #include <assert.h> #include <cmath> // TODO: Define PI #define PI 3.14159 // TODO: Declare abstract class VehicleModel class VehicleModel { // TODO: Declare virtual function Move() virtual void Move(double v, double phi) = 0; }; // TODO: Derive class ParticleModel from VehicleModel class ParticleModel : public VehicleModel { public: // TODO: Override the Move() function void Move(double v, double phi) override { theta += phi; x += v * cos(theta); y += v * sin(theta); } // TODO: Define x, y, and theta double x = 0; double y = 0; double theta = 0; }; // TODO: Derive class BicycleModel from ParticleModel class BicycleModel : public ParticleModel { public: // TODO: Override the Move() function void Move(double v, double phi) override { theta += v / L * tan(phi); x += v * cos(theta); y += v * sin(theta); } // TODO: Define L double L = 1; }; // TODO: Pass the tests int main() { // Test function overriding ParticleModel particle; BicycleModel bicycle; particle.Move(10, PI / 9); bicycle.Move(10, PI / 9); assert(particle.x != bicycle.x); assert(particle.y != bicycle.y); assert(particle.theta != bicycle.theta); }

Compile & Execute Explain

Loading terminal (id_h177dzf), please wait...

Loading [MathJax]/extensions/Safe.js

Menu Shrink

NEXT