

Lesson 2:
Intro to OOP

SEARCH

RESOURCES

CONCEPTS

1. Classes and OOP

2. Bjarne On Classes In C++

3. Jupyter Notebooks

4. Structures

5. Member Initialization

6. Access Specifiers

7. Classes

8. Encapsulation and Abstraction

9. Bjarne on Encapsulation

10. Constructors

11. Scope Resolution

12._INITIALIZER Lists

13. Initializing Constant Members

14. Encapsulation

15. Accessor Functions

16. Mutator Functions

17. Quiz: Classes In C++

18. Exercise: Pyramid Class

19. Exercise: Student Class

20. Encapsulation in C++

21. Bjarne On Abstraction

22. Abstraction

23. Exercise: Sphere Class

24. Exercise: Private Method

25. Exercise: Static Members

26. Exercise: Static Methods

27. Bjarne On Solving Problems

Exercise: Static Methods

SEND FEEDBACK

Exercise: Static Method

In addition to `static` member variables, C++ supports `static` member functions (or "methods"). Just like `static` member variables, `static` member functions are instance-independent: they belong to the class, not to any particular instance of the class.

One corollary to this is that we can method invoke a `static` member function *without ever creating an instance of the class*.

You will try just that in this exercise.

Instructions

1. Refactor `class Sphere` to move the volume calculation into a `static` function.

2. Verify that the class still functions as intended.

3. Make that `static` function public.

4. Call that static function directly from `main()` to calculate the hypothetical volume of a sphere you have not yet instantiated.

https://youtu.be/8ezbkN76msY

Saving Graffiti Recording. Please wait...

Menu

Expand

NEXT