

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

Encapsulation in C++

SEND FEEDBACK

QUESTION 1 OF 3

In the context of object oriented programming, encapsulation refers to:

A requirement that data and logic be packaged separately in distinct objects

The notion that data and logic can be packaged together and passed around within a program as a single object.

The restriction that logic within a particular object can only operate on data stored within that same object.

SUBMIT

QUESTION 2 OF 3

Invoking the `const` keyword in an accessor function allows you to:

Require that the data type of the output will be the same as that of the input.

Ensure the user cannot do anything to change the private attributes of the object.

Pass in constant attribute values to the accessor function.

SUBMIT

QUESTION 3 OF 3

Making class attributes private and assigning them with a mutator function allows you to:

Ensure that only class member functions have access to private class attributes.

Invoke logic that checks whether the input data are valid before setting attributes.

Prevent users from changing non-public class attributes.

SUBMIT

NEXT