

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

Creating A Class

Define your own `Date` class, and provide the proper set of invariants.

```
In [ ]: #include <iostream>
#include <cassert>

class Date {
public:
    int Day() { return day; }
    void Day(int d) {
        if (d >= 1 && d <= 31) day = d;
    }
    int Month() { return month; }
    void Month(int m) {
        if (m >= 1 && m <= 12)
            month = m;
    }
    int Year() { return year; }
    void Year(int y) {
        year = y;
    }

private:
    int day{1};
    int month{1};
    int year{0};
};

// Test in main
int main()
{
    Date date;
    date.Day(-1);
    date.Month(14);
    date.Year(2000);
    assert(date.Day() != -1);
    assert(date.Month() != 14);
    assert(date.Year() == 2000);
    std::cout << date.Day() << "/" << date.Month() << "/" << date.Year() << "\n";
}
```

Compile & Run Explain

Loading terminal (id_8urr4x6), please wait...