

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: Student Class

SEND FEEDBACK

Define a Student class

Follow the instructions to create and test a Student class. If you get stuck you can scroll down to see an example solution.

In []:

```
#include <cassert>
#include <stdexcept>
#include <string>

using std::string;

// TODO: Define "Student" class
class Student {
public:
    // constructor
    Student(string name, int grade, float gpa) : name_(name), grade_(grade), gpa_(gpa) {
        Validate();
    }
    // accessors
    string Name() const {
        return name_;
    }
    int Grade() const {
        return grade_;
    }
    float GPA() const {
        return gpa_;
    }

    // mutators
    void Name(string name) {
        name_ = name;
        Validate();
    }
    void Grade(int grade) {
        grade_ = grade;
        Validate();
    }
    void GPA(float gpa) {
        gpa_ = gpa;
        Validate();
    }

private:
    string name_;
    int grade_;
    float gpa_;
    void Validate() {
        if(Grade() < 0 || Grade() > 12 || GPA() < 0.0 || GPA > 4.0)
            throw std::invalid_argument("argument out of bounds");
    }
};

// TODO: Test
int main() {
    Student david("David Silver", 10, 4.0);
    assert(david.Name() == "David Silver");
    assert(david.Grade() == 10);
    assert(david.GPA() == 4.0);

    bool caught{false};
    try {
        david.Grade(20);
    }
    catch(...) {
        caught = true;
    }
    assert(caught);
}
```

Compile & Run

Explain

Loading terminal (id_un5ro9g), please wait...

↑ Menu

Shrink

NEXT