

☑ 17. Quiz: Classes in C++

18. Exercise: Pyramid Class

19. Exercise: Student Class

🗹 20. Encapsulation in C++

🗹 21. Bjarne On Abstraction

23. Exercise: Sphere Class

24. Exercise: Private Method

25. Exercise: Static Members

26. Exercise: Static Methods

27. Bjarne On Solving Problems

22. Abstraction

Fig. Call Section When Go Doby Tenning Help

Proposed Section Section

Initializer Lists

https://youtu.be/cqKuYu1oiow

SEND FEEDBACK

Initializer Lists

Initializer lists initialize member variables to specific values, just before the class constructor runs. This initialization ensures that class members are automatically initialized when an instance of the class is created.

Date::Date(int day, int month, int year) : year_(y) {
 Day(day);
 Month(month);
}

In this example, the member value year is initialized through the initializer list, while day and month are assigned from within the constructor. Assigning day and month allows us to apply the invariants set in the mutator.

In general, **prefer initialization to assignment**. Initialization sets the value as soon as the object exists, whereas assignment sets the value only after the object comes into being. This means that assignment creates and opportunity to accidentally use a variable before its value is set.

In fact, initialization lists ensure that member variables are initialized *before* the object is created. This is why class member variables can be declared **const**, but only if the member variable is initialized through an initialization list. Trying to initialize a **const** class member within the body of the constructor will not work.

Instructions

Declare class Person.
 Add std::string name to class Person.
 Create a constructor for class Person.
 Add an initializer list to the constructor.

Saving Graffiti Recording. Please wait...

5. Create class object.

↑ Menu 🗾 Expand