Your Box It! submission got 30.00 points. Share Tweet

Try the Next Challenge | Try a Random Challenge

Box It!

by vatsalchanana

Problem Submissions Leaderboard Discussions

Design a class named *Box* whose dimensions are integers and private to the class. The dimensions are labelled: length l, breadth b, and height h.

The default constructor of the class should initialize l, b, and h to 0.

The parameterized constructor *Box(int length, int breadth, int height)* should initialize *Box*'s l, b and h to length, breadth and height.

The copy constructor *Box*(*Box* B) should set l, b and h to B's l, b and h, respectively.

Apart from the above, the class should have 4 functions:

- *int getLength()* - Return box's length
- *int getBreadth()* - Return box's breadth
- *int getHeight()* - Return box's height
- *long long CalculateVolume()* - Return the volume of the box

Overload the operator $<$ for the class *Box*. *Box* $A <$ *Box* B if:

1. $A.l < B.l$
2. $A.b < B.b$ and $A.l == B.l$
3. $A.h < B.h$ and $A.b == B.b$ and $A.l == B.l$

Overload operator $<<$ for the class *Box()*.
If B is an object of class *Box*:

$cout << B$ should print $B.l$, $B.b$ and $B.h$ on a single line separated by spaces.

Constraints

$0 \leq l, b, h \leq 10^5$
Two boxes being compared using the $<$ operator will not have all three dimensions equal.

Easy Submitted 15606 times
Max Score 30

Need Help?

View Discussions

View Top Submissions

Rate This Challenge:

☆☆☆☆☆

Download problem statement

Download sample test cases

Suggest Edits

Current Buffer (saved locally, editable) C++14

```
#include<↔

using namespace std;


#define uint64 unsigned long long
class Box
{
private:
    int m_length;
    int m_breadth;
    int m_height;
public:
    Box()
        :m_length(0), m_breadth(0), m_height(0)
        { }
    Box(const int& l, const int& b, const int& h)
        :m_length(l), m_breadth(b), m_height(h)
        { }
    Box(const Box& B)
        : m_length(B.m_length), m_breadth(B.m_breadth), m_height(B.m_height)
        { }
    const int& getLength()const  {  return m_length;    }
    const int& getBreadth()const {  return m_breadth;   }
    const int& getHeight()const  {  return m_height;    }
    const uint64 CalculateVolume()const  {  return (static_cast<uint64>(m_length)*
                                                    static_cast<uint64>(m_breadth)*
                                                    static_cast<uint64>(m_height)); }

    bool operator< (const Box& obj2)
    {
        return ( (this->m_length < obj2.m_length)                                  ||
        ( (this->m_breadth < obj2.m_breadth) && (this->m_length == obj2.m_length) )||
        ( (this->m_height < obj2.m_height) && (this->m_length == obj2.m_length)
                                        && (this->m_breadth == obj2.m_breadth)) );
    }
    friend std::ostream& operator<< (std::ostream& out, const Box &obj);
};

std::ostream& operator<< (std::ostream& out, const Box &obj)
{
    return out<<obj.m_length<<" "<<obj.m_breadth<<" "<<obj.m_height;
}


void check2()
{↔}

int main()
{↔}
```

Line: 43 Col: 2

Upload Code as File Test against custom input Run Code Submit Code

Congrats, you solved this challenge!

Challenge your friends:

✔ Test Case #0 ✔ Test Case #1 ✔ Test Case #2

✔ Test Case #3 ✔ Test Case #4 ✔ Test Case #5

You've earned 30.00 points. Next Challenge