cs225f20-b.sg

Description | Submission view

Grade

Reviewed on Sunday, 13 September 2020, 12:44 AM by Automatic grade
grade: A

Assessment report[-]
[+]**Summary of tests**

Submitted on Sunday, 13 September 2020, 12:33 AM (Download)

Background.h

```
1  // Provide the interface of the Background class.
2  #include "Color.h"
3
4  #ifndef _BACKGROUND_H_
5  #define _BACKGROUND_H_
6
7  namespace cs225
8  {
9      class Background
10     {
11     private:
12         size_t size;
13         Color* c;
14
15     public:
16         Background();
17         Background(size_t _colourCount, const Color* _colors);
18
19         Background(const Background &rhs);
20         Background &operator= (const Background &rhs);
21
22         const Color operator[](unsigned int i) const;
23
24         size_t count() const;
25
26         ~Background();
27     };
28 }
29
30 #endif
```

Background.cpp

```cpp
// Provide the implementation of the Background class.

#include <iostream>
#include "Background.h"
#include "Color.h"

namespace cs225
{
    Background::Background():size(0),c(new Color[size]){}

    Background::Background(size_t _colourCount, const Color* _colors)
    :size(_colourCount),c(new Color[size])
    {
        for (unsigned int i = 0; i < _colourCount; ++i)
        {
            this->c[i] ={_colors[i].red(),_colors[i].green(),_colors[i].blue()};
        }
    }

    Background::Background(const Background& rhs)
    :size(rhs.size), c(new Color[size])
    {
        for (unsigned int i = 0; i < rhs.size; ++i)
        {
            this->c[i] = { rhs.c[i].red(), rhs.c[i].green(), rhs.c[i].blue() };
        }
    }

    Background &Background::operator=(const Background &rhs)
    {
        if (this == &rhs)
        {
            return*this;
        }

            this->size = rhs.size;
            Color* clr = new Color[rhs.size];

            for (unsigned int i = 0; i < rhs.size; ++i)
            {
                clr[i] = { rhs.c[i].red(), rhs.c[i].green(), rhs.c[i].blue() };
            }

            delete[] this->c;
            this->c = clr;

            return *this;
    }

    const Color Background::operator[](unsigned int i)const
    {
        Color clr = { c[i].red(), c[i].green(), c[i].blue() };
        return clr;
    }

    size_t Background::count() const
    {
        return this->size;
    }

     Background:: ~ Background()
    {
         delete[] c;
    }
}
```

Color.cpp

```cpp
// Provide the implementation of the Color class
// The interface has been already provided for you.

#include "Color.h"

namespace cs225
{
    using type = unsigned char;

    Color::Color(type red, type green, type blue)
    :_red(red), _green(green), _blue(blue){}

    type Color::red() const
    {
        return _red;
    }

    type Color::green() const
    {
        return _green;
    }

    type Color::blue() const
    {
        return _blue;
    }
}
```

VPL

◄ Assignment 1 specification

Jump to...

Attendance cs225f20-b.sg Tuesday 15/9/2020 1:30pm-3:10pm ►