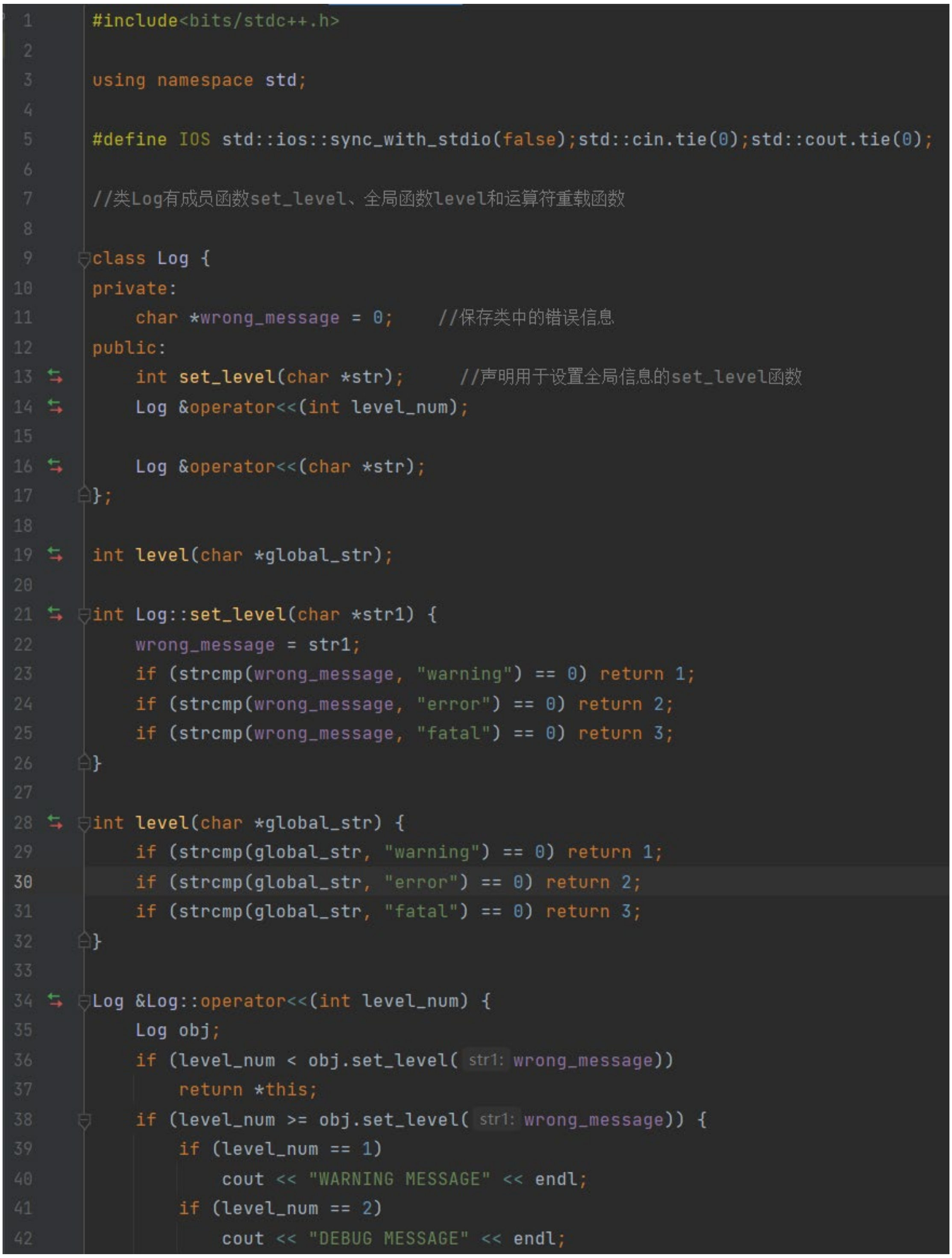
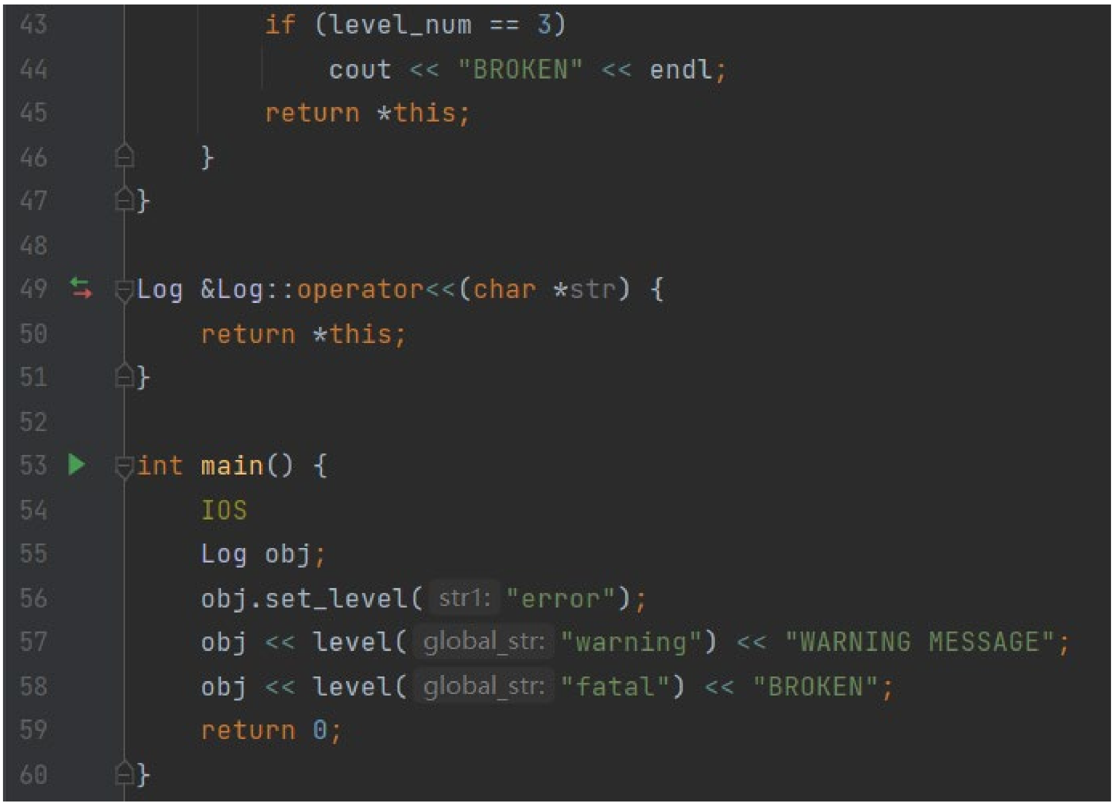
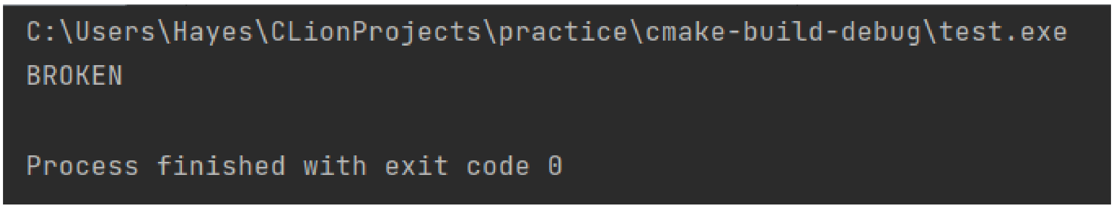


```cpp
#include<bits/stdc++.h>

using namespace std;

#define IOS std::ios::sync_with_stdio(false);std::cin.tie(0);std::cout.tie(0);

//类Log有成员函数set_level、全局函数level和运算符重载函数

class Log {
private:
    char *wrong_message = 0;    //保存类中的错误信息
public:
    int set_level(char *str);     //声明用于设置全局信息的set_level函数
    Log &operator<<(int level_num);

    Log &operator<<(char *str);
};

int level(char *global_str);

int Log::set_level(char *str1) {
    wrong_message = str1;
    if (strcmp(wrong_message, "warning") == 0) return 1;
    if (strcmp(wrong_message, "error") == 0) return 2;
    if (strcmp(wrong_message, "fatal") == 0) return 3;
}

int level(char *global_str) {
    if (strcmp(global_str, "warning") == 0) return 1;
    if (strcmp(global_str, "error") == 0) return 2;
    if (strcmp(global_str, "fatal") == 0) return 3;
}

Log &Log::operator<<(int level_num) {
    Log obj;
    if (level_num < obj.set_level( str1: wrong_message))
        return *this;
    if (level_num >= obj.set_level( str1: wrong_message)) {
        if (level_num == 1)
            cout << "WARNING MESSAGE" << endl;
        if (level_num == 2)
            cout << "DEBUG MESSAGE" << endl;
```



```
        if (level_num == 3)
            cout << "BROKEN" << endl;
        return *this;
    }
}

Log &Log::operator<<(char *str) {
    return *this;
}

int main() {
    IOS
    Log obj;
    obj.set_level( str1: "error");
    obj << level( global_str: "warning") << "WARNING MESSAGE";
    obj << level( global_str: "fatal") << "BROKEN";
    return 0;
}
```

运行结果：



```
C:\Users\Hayes\CLionProjects\practice\cmake-build-debug\test.exe
BROKEN

Process finished with exit code 0
```