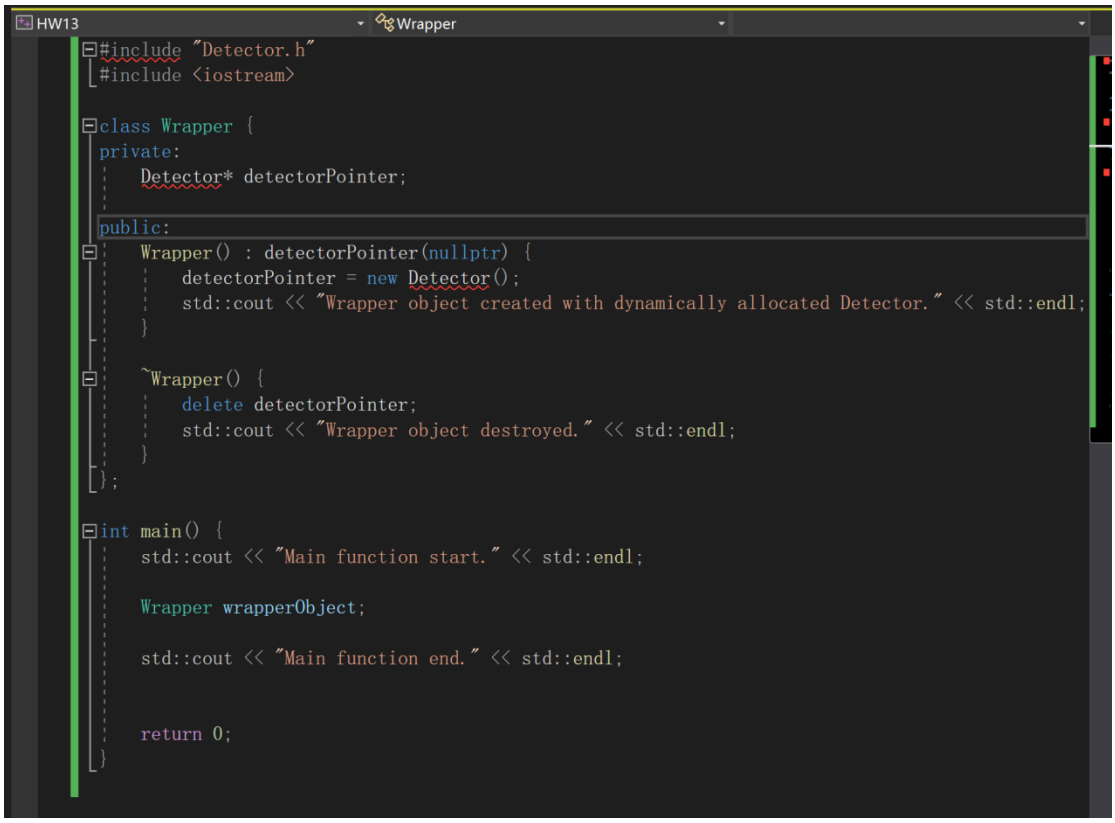
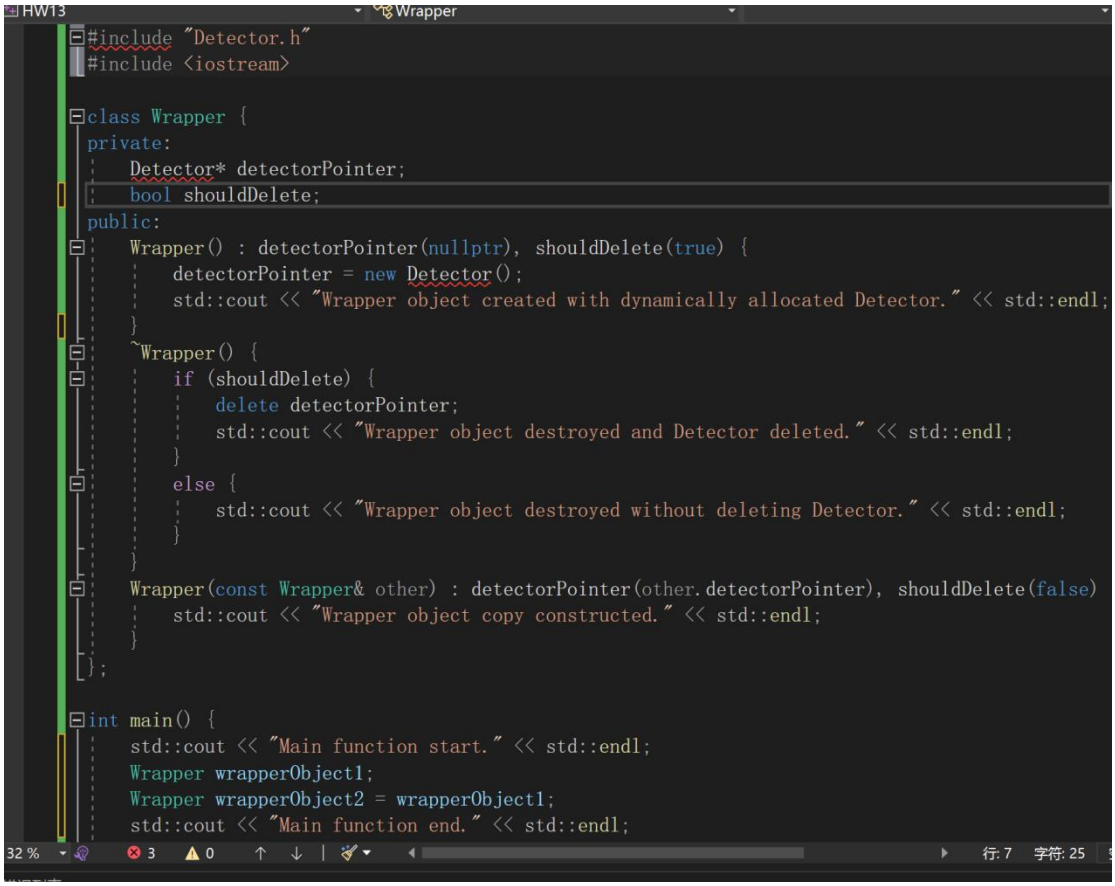
HW13 (全局范围)

```cpp
#include "Detector.h"
#include <iostream>

class Wrapper {
private:
    Detector detectorObject;

public:

    Wrapper() {
        std::cout << "Wrapper object created." << std::endl;
    }

    ~Wrapper() {
        std::cout << "Wrapper object destroyed." << std::endl;
    }
};

int main() {
    std::cout << "Main function start." << std::endl;
    Wrapper wrapperObject;

    std::cout << "Main function end." << std::endl;


    return 0;
}
```

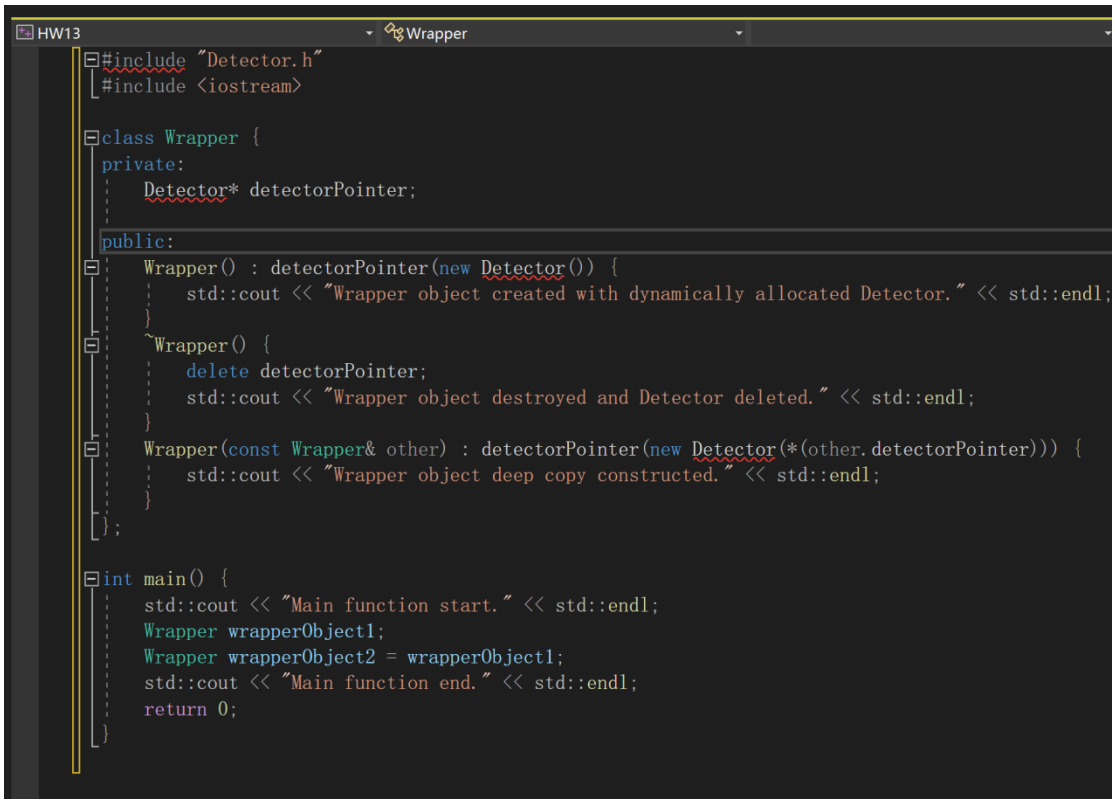
```cpp
class Wrapper {
private:
    Detector* detectorPointer;

public:
    Wrapper() : detectorPointer(nullptr) {
        detectorPointer = new Detector();
        std::cout << "Wrapper object created with dynamically allocated Detector." <
    }

    ~Wrapper() {
        delete detectorPointer;
        std::cout << "Wrapper object destroyed." << std::endl;
    }
};

int main() {
    std::cout << "Main function start." << std::endl;

    Wrapper wrapperObject;

    std::cout << "Main function end." << std::endl;


    return 0;
}
```



```
HW13
Wrapper
#include "Detector.h"
#include <iostream>

class Wrapper {
private:
    Detector* detectorPointer;

public:
    Wrapper() : detectorPointer(nullptr) {
        detectorPointer = new Detector();
        std::cout << "Wrapper object created with dynamically allocated Detector." << std::endl;
    }

    ~Wrapper() {
        delete detectorPointer;
        std::cout << "Wrapper object destroyed." << std::endl;
    }
};

int main() {
    std::cout << "Main function start." << std::endl;

    Wrapper wrapperObject;

    std::cout << "Main function end." << std::endl;


    return 0;
}
```



```
HW13
Wrapper
#include "Detector.h"
#include <iostream>

class Wrapper {
private:
    Detector* detectorPointer;
    bool shouldDelete;
public:
    Wrapper() : detectorPointer(nullptr), shouldDelete(true) {
        detectorPointer = new Detector();
        std::cout << "Wrapper object created with dynamically allocated Detector." << std::endl;
    }
    ~Wrapper() {
        if (shouldDelete) {
            delete detectorPointer;
            std::cout << "Wrapper object destroyed and Detector deleted." << std::endl;
        }
        else {
            std::cout << "Wrapper object destroyed without deleting Detector." << std::endl;
        }
    }
    Wrapper(const Wrapper& other) : detectorPointer(other.detectorPointer), shouldDelete(false)
        std::cout << "Wrapper object copy constructed." << std::endl;
    }
};

int main() {
    std::cout << "Main function start." << std::endl;
    Wrapper wrapperObject1;
    Wrapper wrapperObject2 = wrapperObject1;
    std::cout << "Main function end." << std::endl;
32 %   3   0                                                    行: 7   字符: 25
```


HW13
Wrapper
#include "Detector.h"
#include <iostream>
class Wrapper {
private:
Detector* detectorPointer;
public:
Wrapper() : detectorPointer(new Detector()) {
std::cout << "Wrapper object created with dynamically allocated Detector." << std::endl;
}
~Wrapper() {
delete detectorPointer;
std::cout << "Wrapper object destroyed and Detector deleted." << std::endl;
}
Wrapper(const Wrapper& other) : detectorPointer(new Detector(*(other.detectorPointer))) {
std::cout << "Wrapper object deep copy constructed." << std::endl;
}
};
int main() {
std::cout << "Main function start." << std::endl;
Wrapper wrapperObject1;
Wrapper wrapperObject2 = wrapperObject1;
std::cout << "Main function end." << std::endl;
return 0;
}