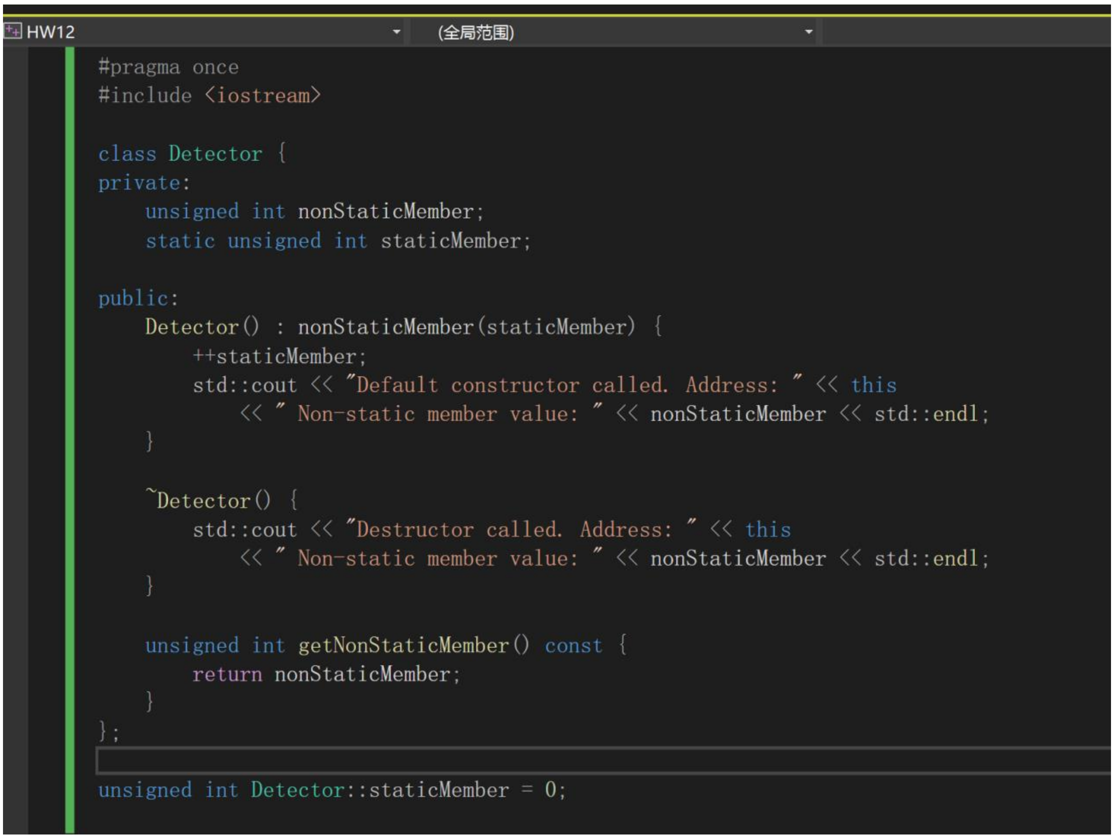
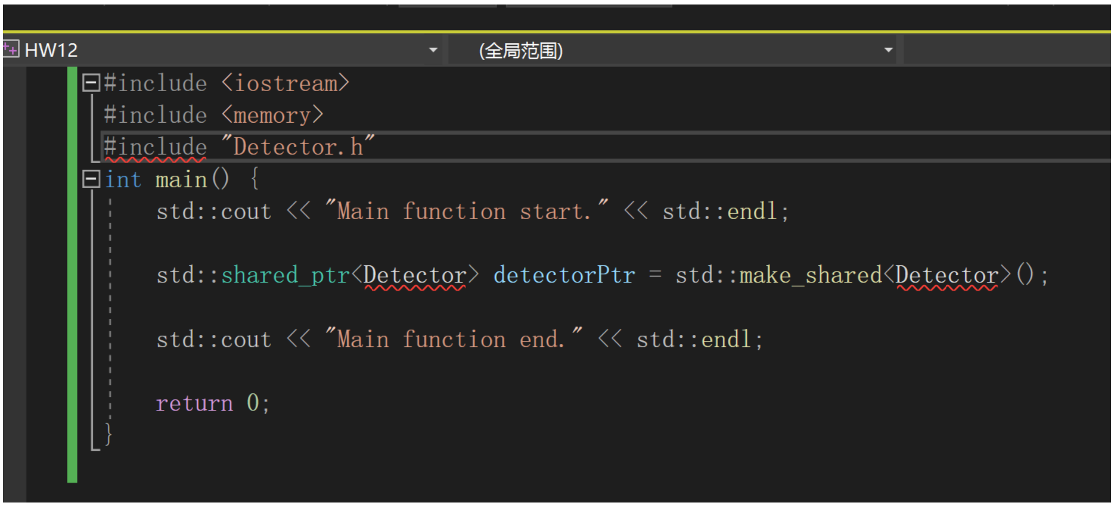
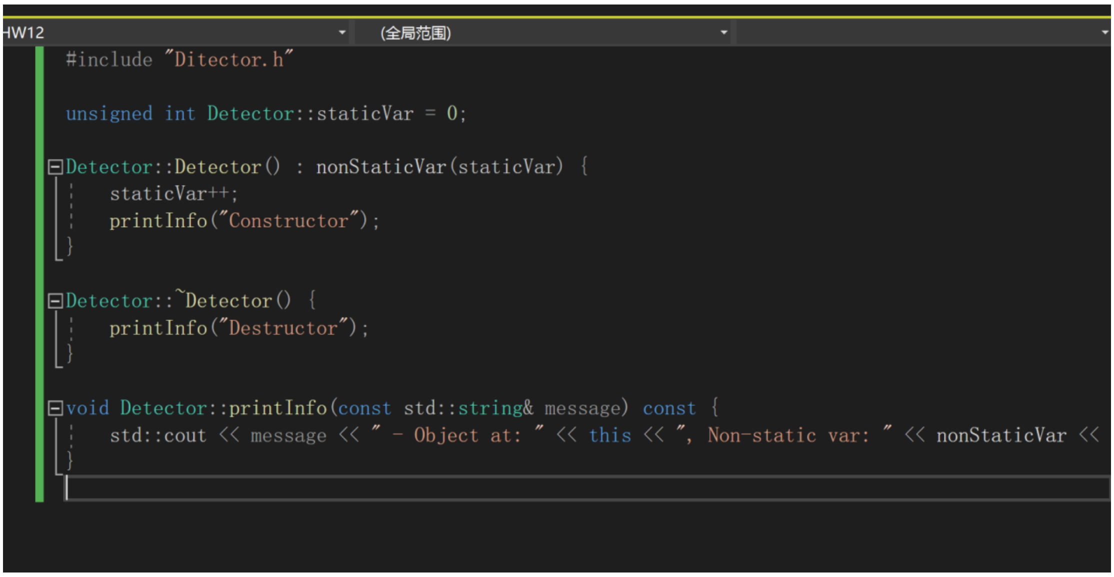


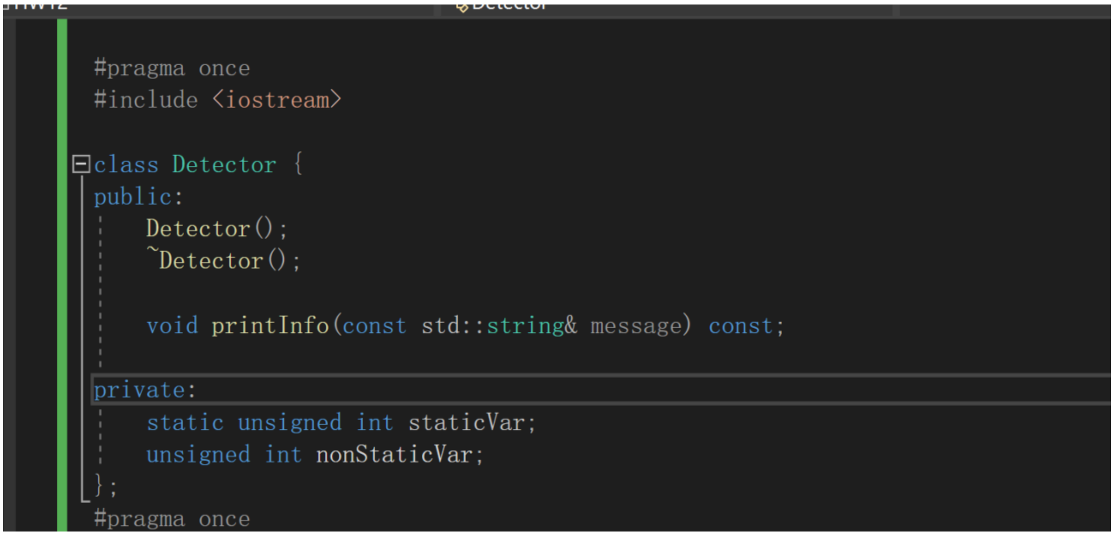
```cpp
#pragma once
#include <iostream>

class Detector {
private:
    unsigned int nonStaticMember;
    static unsigned int staticMember;

public:
    Detector() : nonStaticMember(staticMember) {
        ++staticMember;
        std::cout << "Default constructor called. Address: " << this
            << " Non-static member value: " << nonStaticMember << std::endl;
    }

    ~Detector() {
        std::cout << "Destructor called. Address: " << this
            << " Non-static member value: " << nonStaticMember << std::endl;
    }

    unsigned int getNonStaticMember() const {
        return nonStaticMember;
    }
};

unsigned int Detector::staticMember = 0;
```



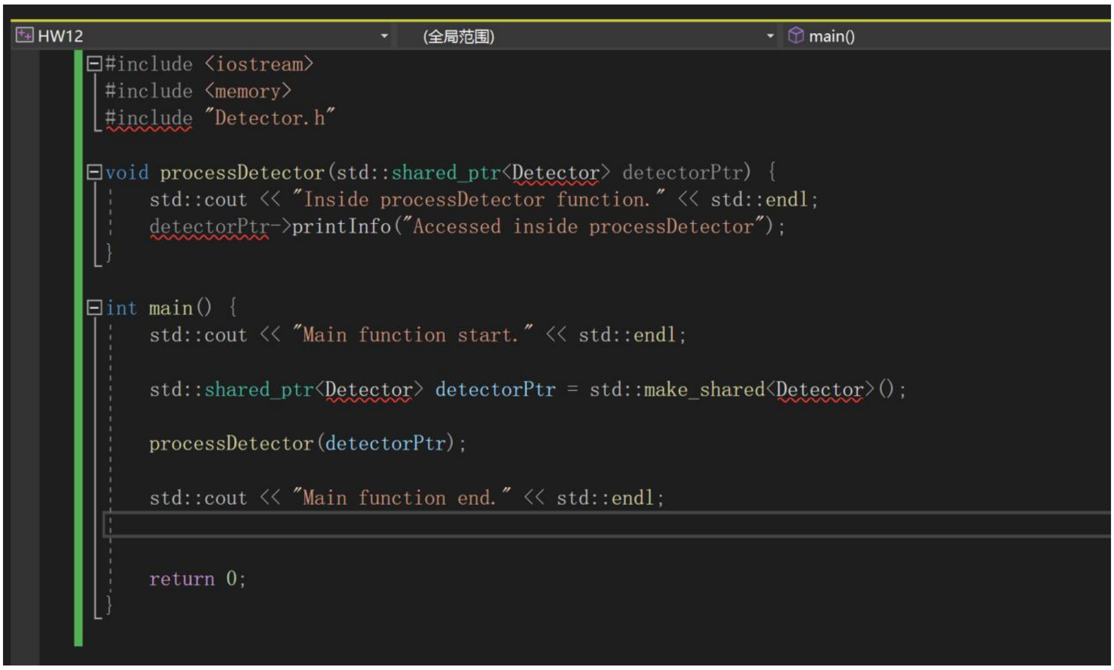
```cpp
#include <iostream>
#include <memory>
#include "Detector.h"
int main() {
    std::cout << "Main function start." << std::endl;

    std::shared_ptr<Detector> detectorPtr = std::make_shared<Detector>();

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

    return 0;
}
```



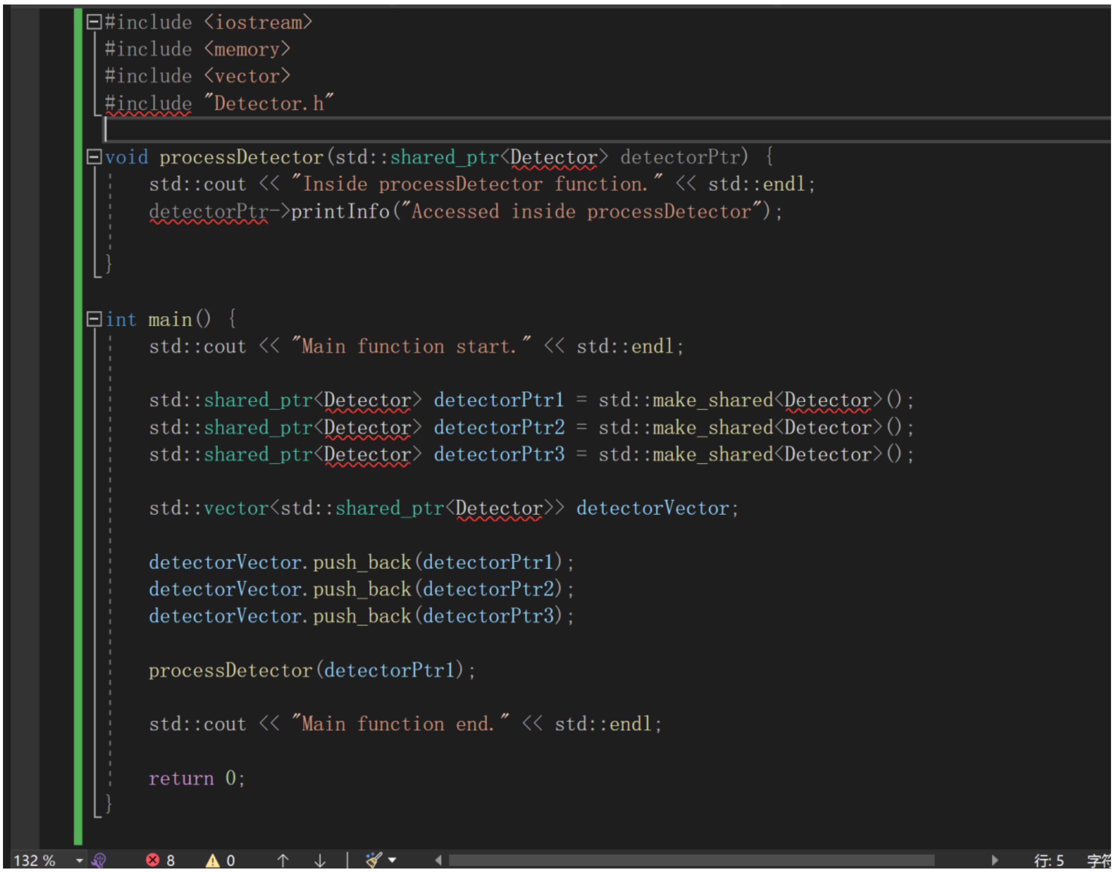
```
#include "Ditector.h"

unsigned int Detector::staticVar = 0;

Detector::Detector() : nonStaticVar(staticVar) {
    staticVar++;
    printInfo("Constructor");
}

Detector::~Detector() {
    printInfo("Destructor");
}

void Detector::printInfo(const std::string& message) const {
    std::cout << message << " - Object at: " << this << ", Non-static var: " << nonStaticVar <<
}
```



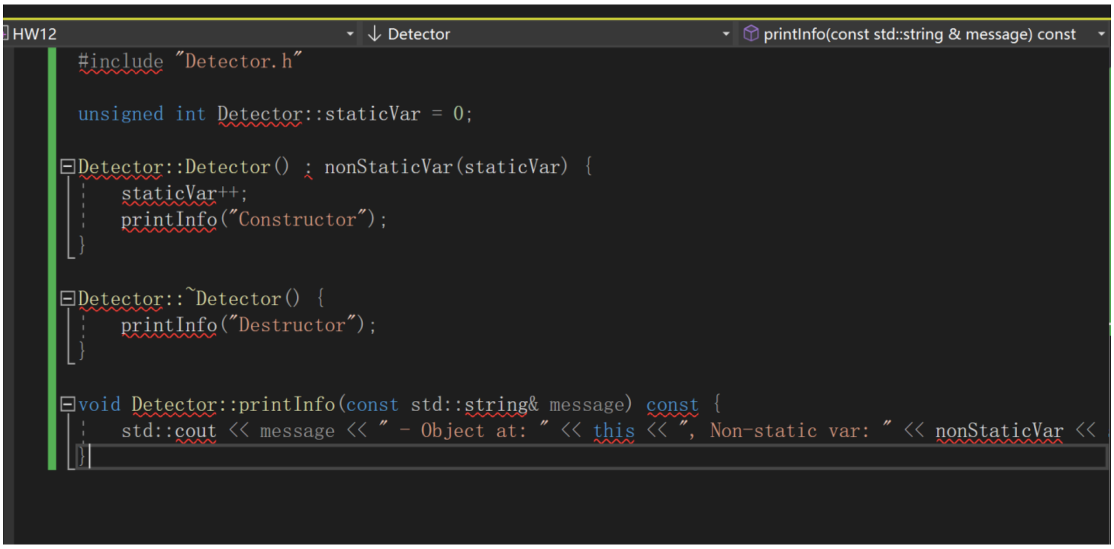
```
#pragma once
#include <iostream>

class Detector {
public:
    Detector();
    ~Detector();

    void printInfo(const std::string& message) const;

private:
    static unsigned int staticVar;
    unsigned int nonStaticVar;
};
#pragma once
```



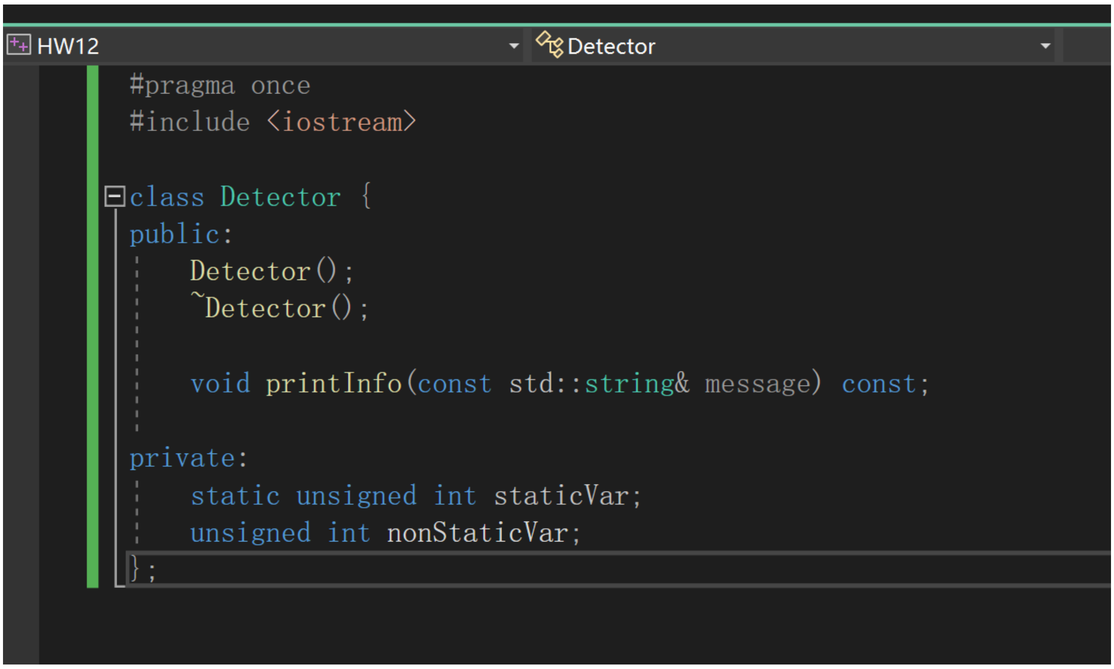
```
#include <iostream>
#include <memory>
#include "Detector.h"

void processDetector(std::shared_ptr<Detector> detectorPtr) {
    std::cout << "Inside processDetector function." << std::endl;
    detectorPtr->printInfo("Accessed inside processDetector");
}

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

    std::shared_ptr<Detector> detectorPtr = std::make_shared<Detector>();

    processDetector(detectorPtr);

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


    return 0;
}
```



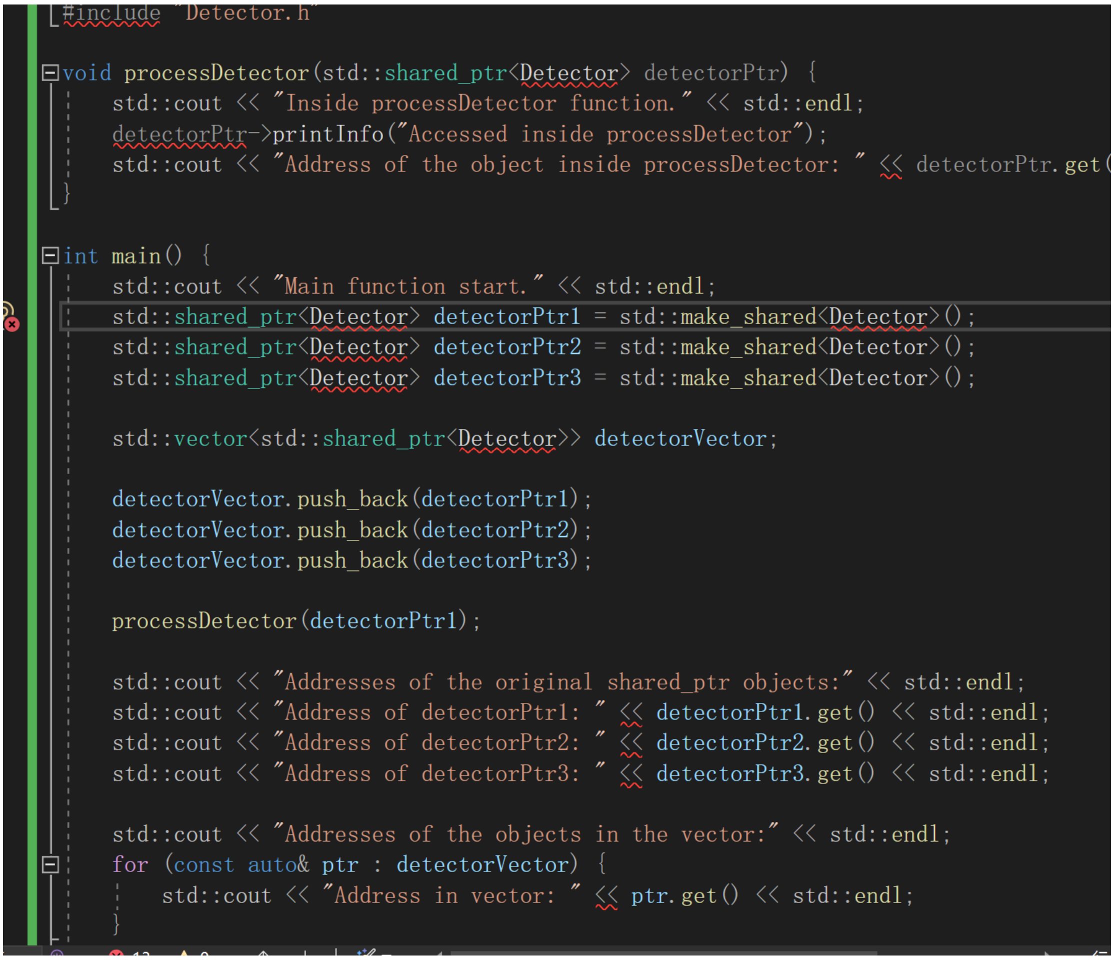
```cpp
#include <iostream>
#include <memory>
#include <vector>
#include "Detector.h"

void processDetector(std::shared_ptr<Detector> detectorPtr) {
    std::cout << "Inside processDetector function." << std::endl;
    detectorPtr->printInfo("Accessed inside processDetector");

}

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

    std::shared_ptr<Detector> detectorPtr1 = std::make_shared<Detector>();
    std::shared_ptr<Detector> detectorPtr2 = std::make_shared<Detector>();
    std::shared_ptr<Detector> detectorPtr3 = std::make_shared<Detector>();

    std::vector<std::shared_ptr<Detector>> detectorVector;

    detectorVector.push_back(detectorPtr1);
    detectorVector.push_back(detectorPtr2);
    detectorVector.push_back(detectorPtr3);

    processDetector(detectorPtr1);

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

    return 0;
}
```



```cpp
#include "Detector.h"

unsigned int Detector::staticVar = 0;

Detector::Detector() : nonStaticVar(staticVar) {
    staticVar++;
    printInfo("Constructor");
}

Detector::~Detector() {
    printInfo("Destructor");
}

void Detector::printInfo(const std::string& message) const {
    std::cout << message << " - Object at: " << this << ", Non-static var: " << nonStaticVar <<
}
```



```cpp
#pragma once
#include <iostream>

class Detector {
public:
    Detector();
    ~Detector();

    void printInfo(const std::string& message) const;

private:
    static unsigned int staticVar;
    unsigned int nonStaticVar;
};
```



```cpp
#include "Detector.h"

void processDetector(std::shared_ptr<Detector> detectorPtr) {
    std::cout << "Inside processDetector function." << std::endl;
    detectorPtr->printInfo("Accessed inside processDetector");
    std::cout << "Address of the object inside processDetector: " << detectorPtr.get(
}

int main() {
    std::cout << "Main function start." << std::endl;
    std::shared_ptr<Detector> detectorPtr1 = std::make_shared<Detector>();
    std::shared_ptr<Detector> detectorPtr2 = std::make_shared<Detector>();
    std::shared_ptr<Detector> detectorPtr3 = std::make_shared<Detector>();

    std::vector<std::shared_ptr<Detector>> detectorVector;

    detectorVector.push_back(detectorPtr1);
    detectorVector.push_back(detectorPtr2);
    detectorVector.push_back(detectorPtr3);

    processDetector(detectorPtr1);

    std::cout << "Addresses of the original shared_ptr objects:" << std::endl;
    std::cout << "Address of detectorPtr1: " << detectorPtr1.get() << std::endl;
    std::cout << "Address of detectorPtr2: " << detectorPtr2.get() << std::endl;
    std::cout << "Address of detectorPtr3: " << detectorPtr3.get() << std::endl;

    std::cout << "Addresses of the objects in the vector:" << std::endl;
    for (const auto& ptr : detectorVector) {
        std::cout << "Address in vector: " << ptr.get() << std::endl;
    }
```