

```
#include <iostream>
#include <memory>
#include <string>

using namespace std;

class Student {
private:
    int id;
    string first;
    string last;

public:
    Student(int id, const string& first, const string& last) :
        id(id), first(first), last(last) {}

    Student() :
        id{}, first({}), last({}) {}

    int getId() {
        return id;
    }

    const string& getFirst() const {
        return first;
    }

    const string& getLast() const {
        return last;
    }
};

class StudentDeleter {
public:
    void operator()(Student* s) const {
        cout << "StudentDeleter: deleting " << s->getFirst()
            << " " << s->getLast() << "\n";
        delete s;
    }
};

void printStudent(weak_ptr<Student> s) {
    cout << s.use_count() << " references open\n";
    if (shared_ptr<Student> l = s.lock()) {
        cout << l->getFirst() << " " << l->getLast() << "\n";
    }
    else {
        cout << "pointer has expired\n";
    }
}

void printSharedStudent(shared_ptr<Student> s) {
    cout << s.use_count() << " references open\n";
    cout << s->getFirst() << " " << s->getLast() << "\n";
}

int main() {
    weak_ptr<Student> w;

    {
        shared_ptr<Student> s(new Student(1, "John", "Smith"));

        StudentDeleter sd;
        Student* j = new Student(2, "James", "Slocum");

        shared_ptr<Student> student(j, sd);

        printStudent(s);
    }
}
```

```
    printStudent(student);  
  
    printSharedStudent(s);  
    printSharedStudent(student);  
  
    w = student;  
    printStudent(w);  
}  
  
printStudent(w);  
  
return 0;  
}
```