Verify your understanding!!!

• Get into groups of 3-4.
• Mark down line # where the library default constructor, copy constructor, assignment operator overload, and destructor are called
• Mark down the number of times the library default constructor, copy constructor, assignment operator overload, and destructor are called
• When is patron non-default constructor and destructor are called?
• When is patron copy constructor and assign op overload called?
#include "library.h"
#include <iostream>
using namespace std;

void fun(library l) {
    cout << l.get_patron()->get_name() << endl;
}

library fun() {
    library lib;
    return lib;
}

library fun_ref(library &l) {
    library lib;
    l=lib;
    return lib;
}

void fun(library *l) {
    library lib=*l;
}

int main() {
    library l;
    library l2=fun_ref(l);
    library l3(l2);
    fun(&l3);
    l2=fun_ref(l);
    l3=l2;
    library l4=fun();
    fun(l4);
    return 0;
}
What is called in patron class?

```cpp
#include <iostream>

class Library {
    std::string name;
    int id;
    Library(const Library &l) : p(l.p) {
        cout << "Library object copy constructor!" << endl;
    }
    void Library::operator=(const Library &l) {
        p=l.p;
        cout << "Library object assignment overload!" << endl;
    }
    Library() {
        cout << "Library object being destructed!" << endl;
    }

private:
    int p;
};

int main() {
    Library lib1; // Create a Library object
    Library lib2 = lib1; // Copy constructor
    lib2 = lib1; // Assignment operator
    return 0;
}
```

Is this allowable without patron default constructor?
Patron constructors, destructor, op overload

```cpp
// once you make a non-default constructor, then you
// do not have a default constructor implicitly defined
patron::patron(string s) {
    name = s;
    cout << "in patron non-default constructor" << endl;
}
patron::patron(const patron &p) : name(p.name) {
    cout << "in patron copy constructor" << endl;
}
patron::~patron() {
    cout << "in patron destructor" << endl;
}
void patron::operator =(const patron &p) {
    name = p.name;
    cout << "in patron assign op overload" << endl;
}
```