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?
```cpp
#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
library::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::~library() {
    cout << "Library object being destructed!" << endl;
}
```
Is this allowable without patron default constructor?

```
library::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::~library() {
    cout << "Library object being destructed!" << endl;
}
```
Patron constructors, destructor, op overload

```cpp
3 // once you make a non-default constructor, then you
4 // do not have a default constructor implicitly defined
5 patron::patron(string s) {
   6     name=s;
   7     cout << "in patron non-default constructor" << endl;
   8 }
9 patron::patron(const patron &p) : name(p.name) {
   10     cout << "in patron copy constructor" << endl;
   11 }
12 patron::~patron() {
   13     cout << "in patron destructor" << endl;
   14 }
15 void patron::operator=(const patron &p) {
   16     name=p.name;
   17     cout << "in patron assign op overload" << endl;
   18 }```