OBJECT CLONING

Emilie Menard Barnard
Foothill College
April 13, 2017

OUTLINE

- Pointers Review
- Object Cloning
  - Copy constructor
  - Shallow vs. deep copy
  - Use case diagram examples

POINTERS REVIEW

- A **Pointer** contains an address which *points* to another location in memory

- In C++:

```
addressOfmyVar = &myVar;
valueOfmyVar = *addressOfmyVar;
myObjectMember = myObject.member;
myMemberOfPointerObject = (*myPointer).member;
myMemberOfPointerObject = myPointer -> member;
```

OBJECT CLONING

- Objects are copied when:
  - A class object is declared and initialized by another object of the same type
  - A function returns a value of the class type
  - An argument of the class type is "plugged in" for a call-by-value parameter

OBJECT CLONING

- A **copy constructor** is used to create a new instance of the same object. One of its parameters must be the same type of the object's class.

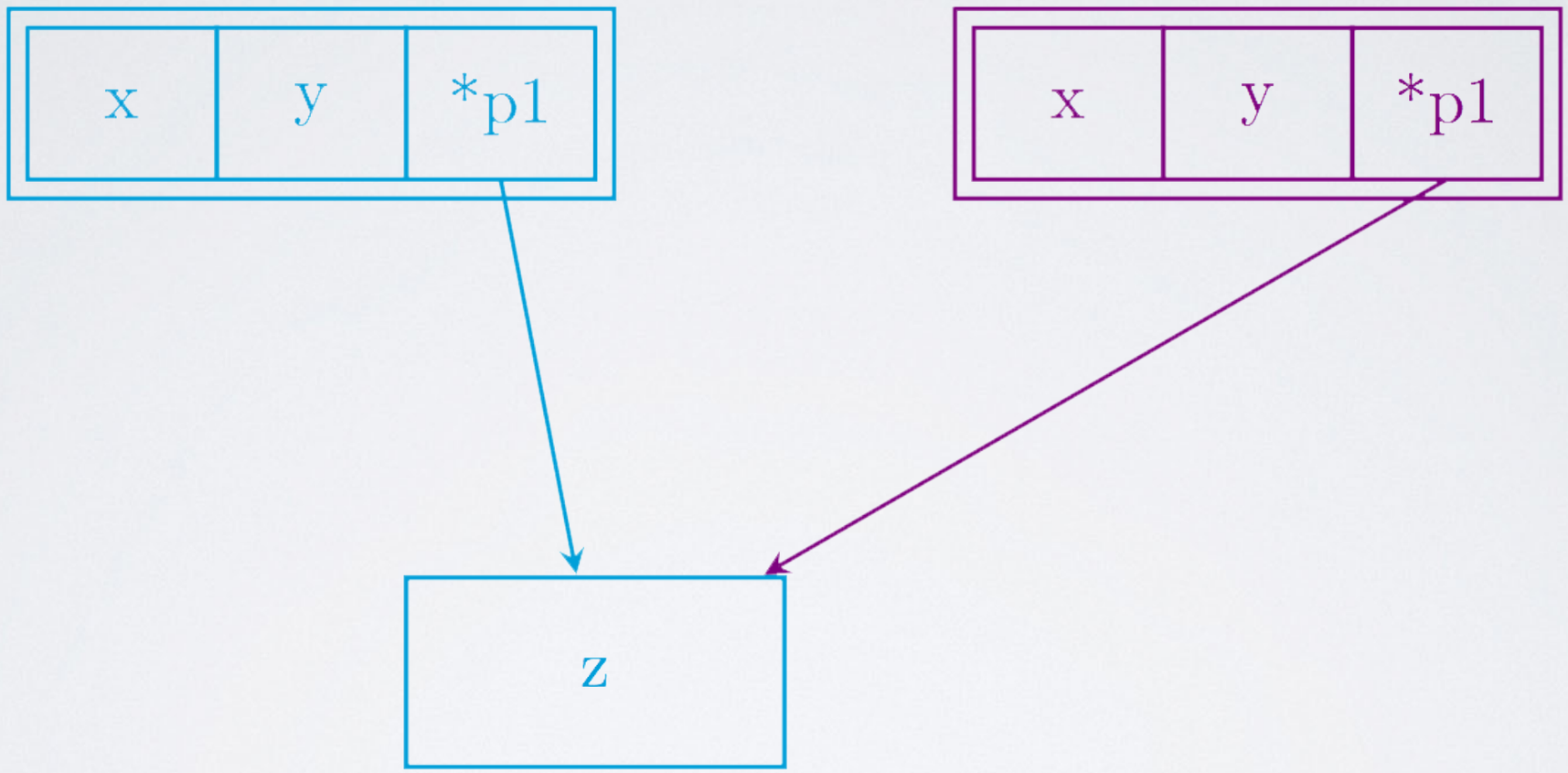
- C++ has a default copy constructor if you do not make your own.

- There are cases where you will want to write your own.

SHALLOW COPY

- New object is created.
- Member variables are copied over to the new object.
  - If any happen to be pointers, the pointer itself is copied, creating another pointer to its object.
- *Shallow*: everything at the "surface" level is copied over to the new object
- The default C++ copy constructor creates a shallow copy

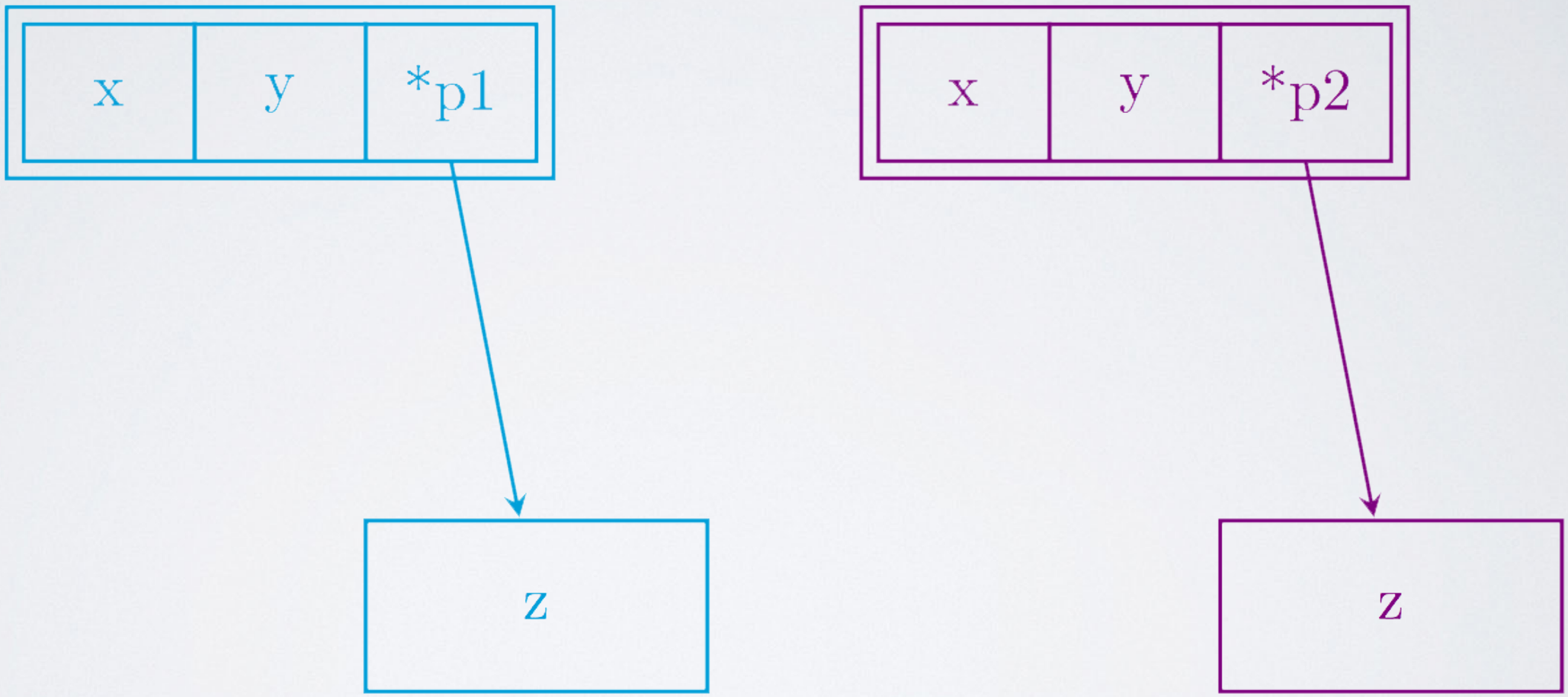
SHALLOW COPY EXAMPLE



DEEP COPY

- New object is created.
- Member variables are copied into the new object.
  - If any happen to be pointers, the pointer's object is duplicated and a new pointer to this copied object is created.
- *Deep*: dive down as far as you can in the object, and copy everything

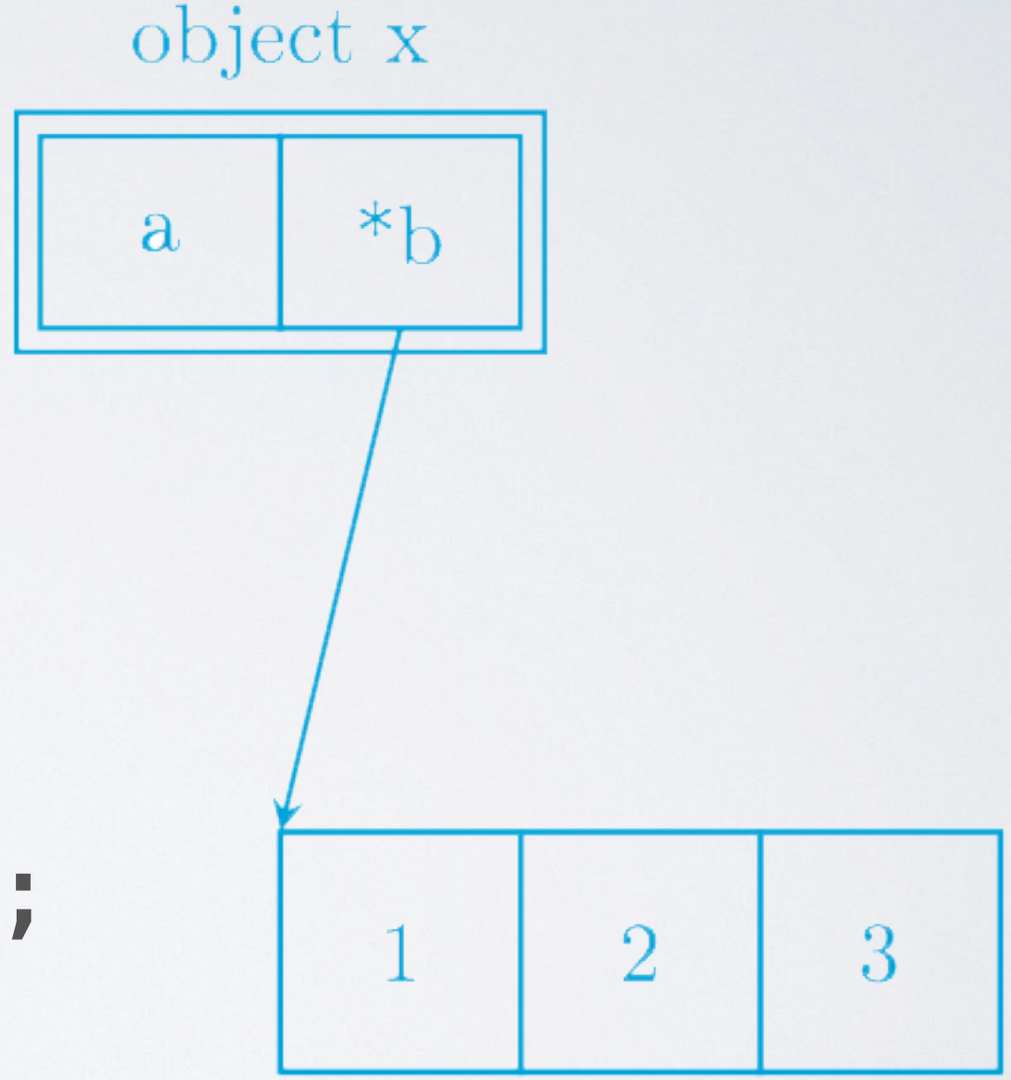
DEEP COPY EXAMPLE



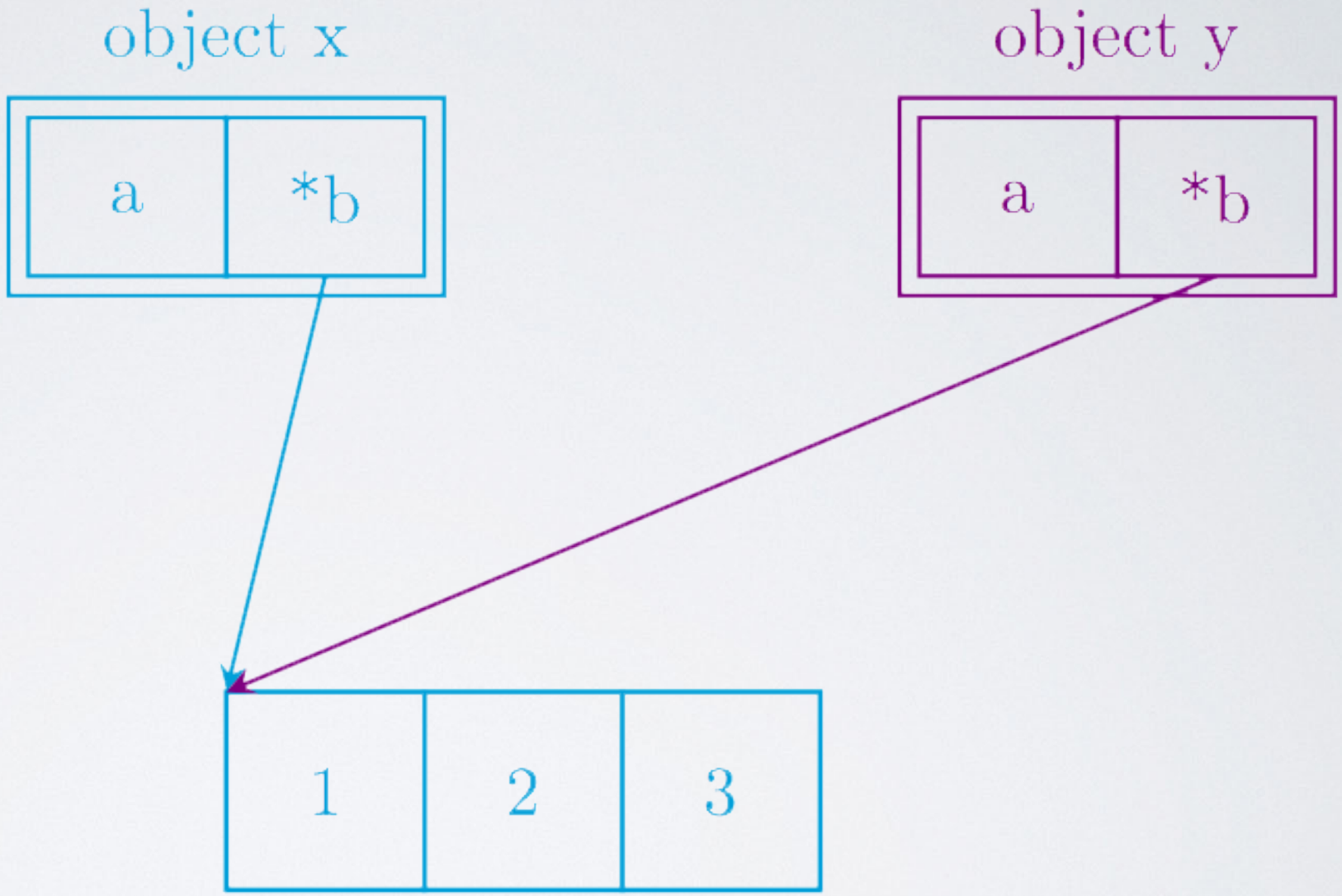
ANOTHER EXAMPLE

What will print when we make a *shallow* copy y of object x and run the following code?

```
y.b[0] = 5;
cout << x.b[0] << endl;
```

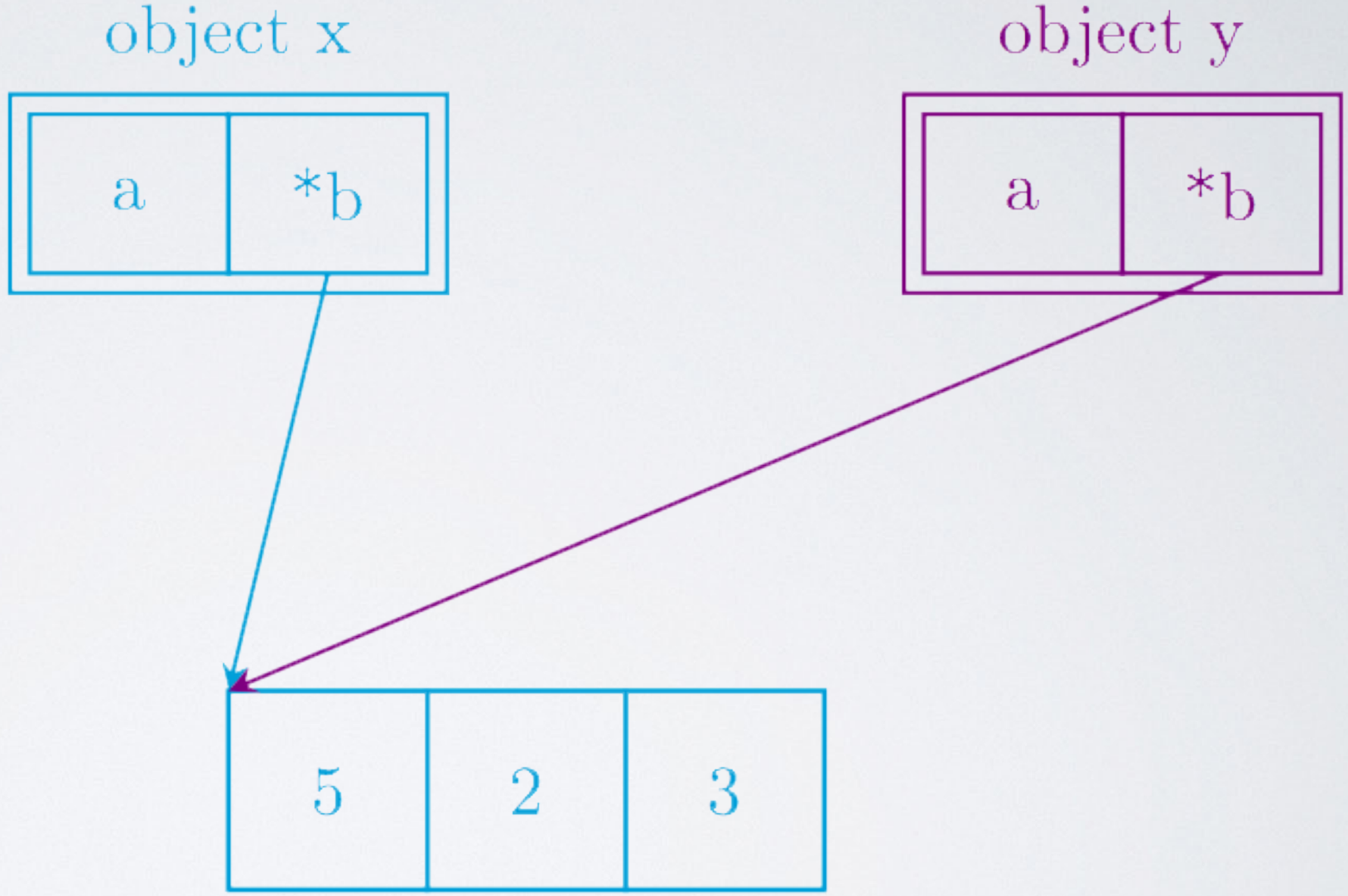


SHALLOW COPY




```
y.b[0] = 5;
cout << x.b[0] << endl;
```

SHALLOW COPY



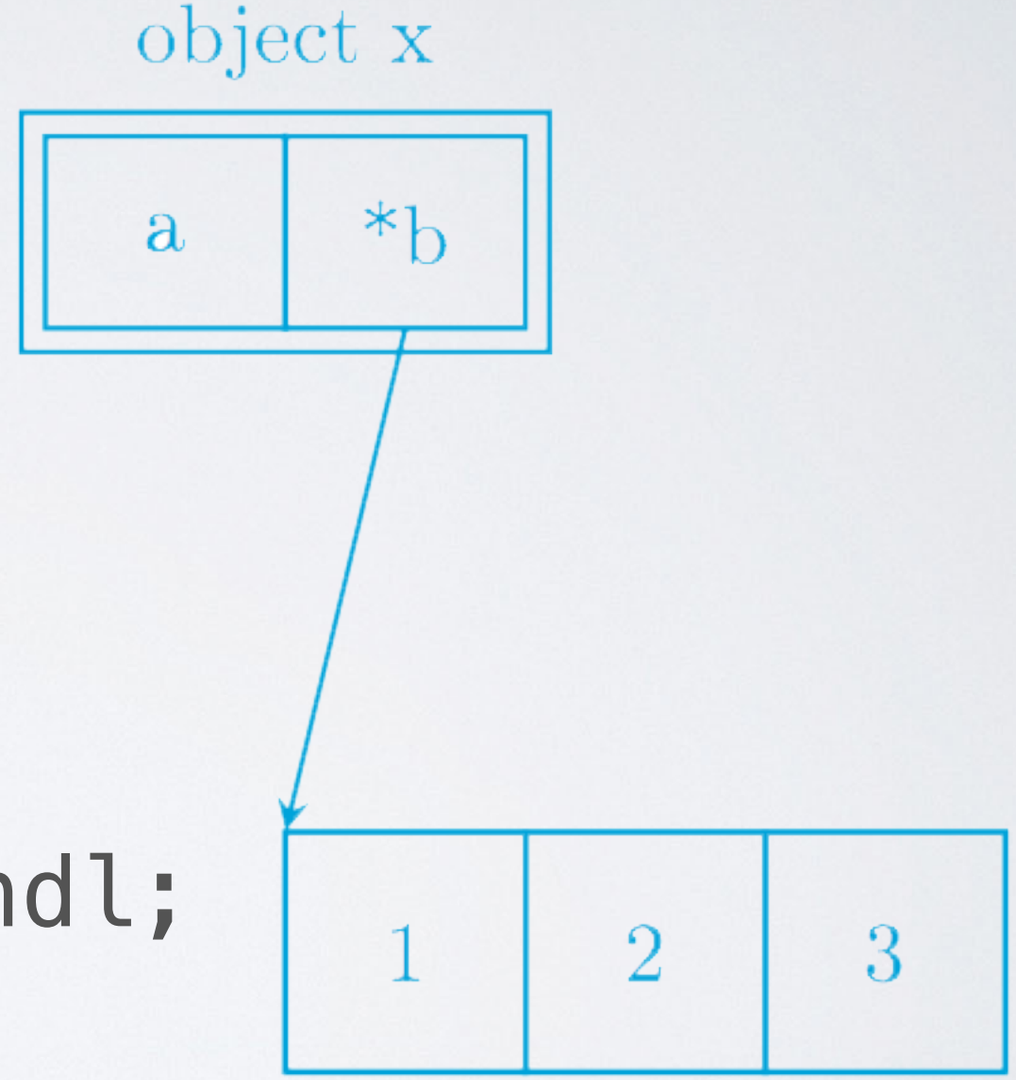

```
y.b[0] = 5;
cout << x.b[0] << endl;
```

Prints: 5

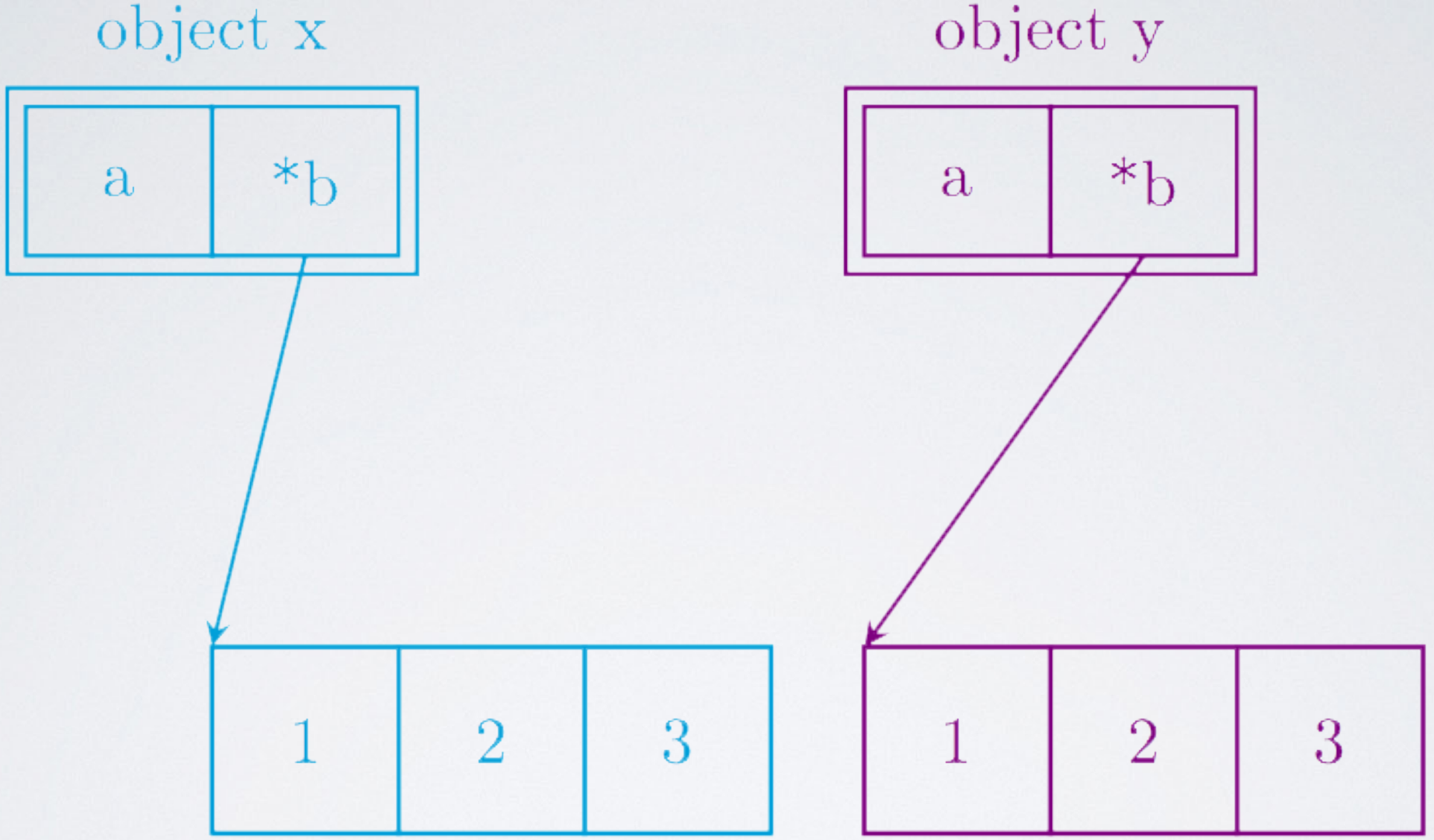
ANOTHER EXAMPLE

What will print when we make a *deep* copy y of object x and run the following code?

```
y.b[0] = 5;
cout << x.b[0] << endl;
```

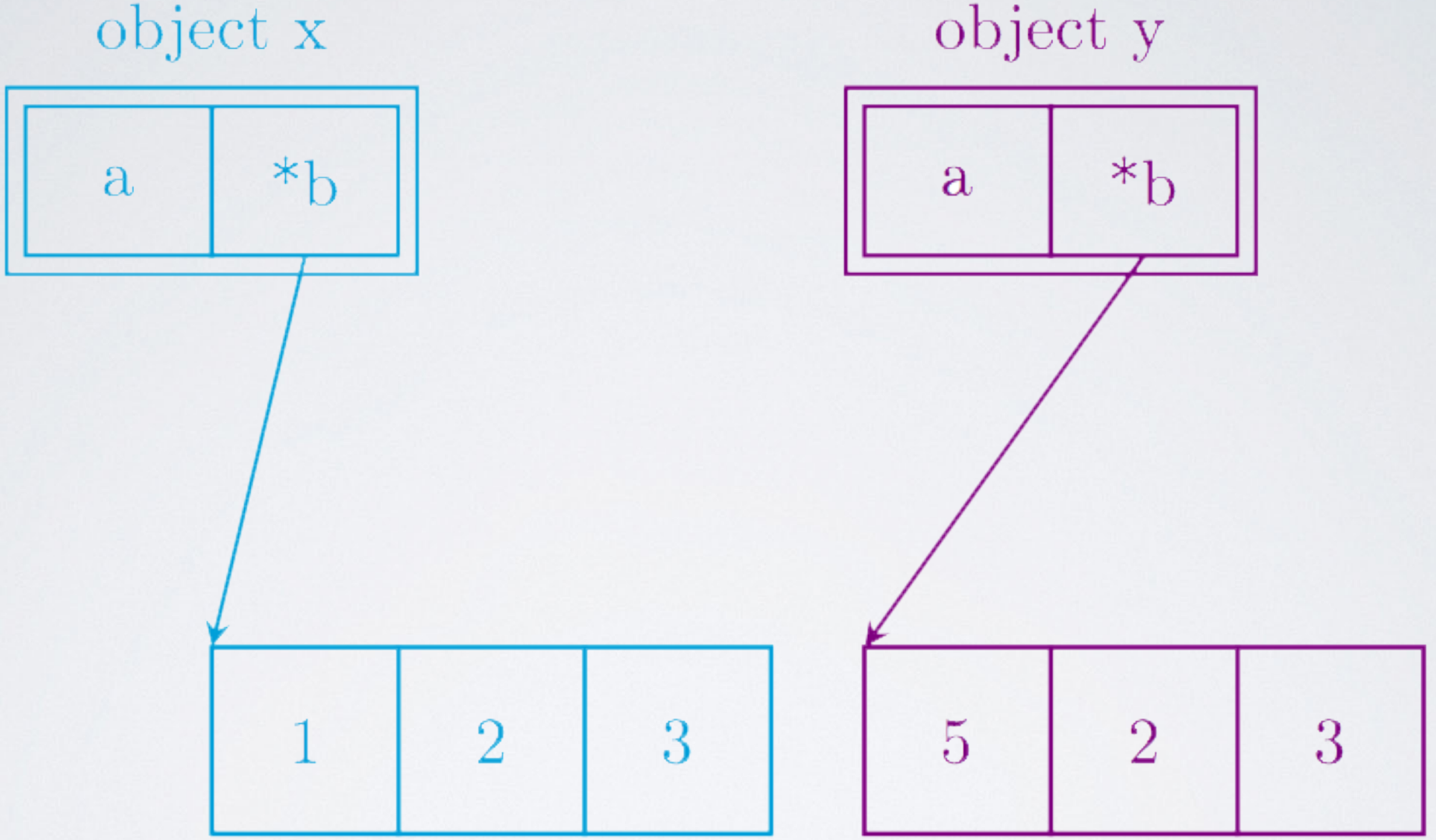


DEEP COPY




```
y.b[0] = 5;
cout << x.b[0] << endl;
```

DEEP COPY




```
y.b[0] = 5;
cout << x.b[0] << endl;
```

Prints: 1

READING & RESOURCES

- Absolute C++: pages 445 - 450

- Code examples posted on website:
  - Default copy constructor
  - Defining your own copy constructor
  - Lecture example of shallow vs. deep copies