Office Hour #1

constructors,destructors

```
class A {
  public:
    A() {cout << "A";}
    ~A() {cout << "~A";}
};

class B {
  public:
    B() {cout << "B";}
    ~B() {cout << "~B";}
  private:
    A a[2];
}

void foo(B b) {                     // 2. AA
  A* aa;
  A a2;
}

int main() {
  B b;                              // 1. AAB
  foo(b);
}

// What is the output?

// AAB | A   ~A~B~A~A | ~B~A~A
// AAB | AABA~A~B~A~A | ~B~A~A
```

- member variables are constructed first in a class declaration, before the actual constructor runs
- destruction order is reverse construction order **(first in, last out)**
    - **AB~B~A**
- if there are no copy constructors called in this situation, the construction would be a palindrome of the destruction
    - AAB ~B~A~A

```
class Foo {
  public:
    Foo() {ptr = nullptr;}
    Foo& operator=(const Foo &other) {
      if (this == &other) return *this;
      foo temp(other);
      swap(this->ptr, temp->ptr);
      return *this;
    }
    ~Foo() {delete ptr;}

    void createNerd() {
```

```
    delete ptr;
    ptr = new Nerd;
  }

private:
  Nerd *ptr;
};

int main() {
  foo a;
  a.createNerd();
  foo b;
  b = a;
}

```