

```
class ECE462XX {
public:
    ECE462XX() {
        size = 10;
        data = new [size];
    }
}
```

```
ECE462XX(const ECE462XX& orig) {
    size = orig.size;
    data = new [size];
    for (int i = 0; i < size; i++) {
        data[i] = orig.data[i];
    }
}
```

↳ Copy constructor
Compiler provides shallow copy

//overrides
Equal sign

```
ECE462XX& operator=(const ECE462XX& orig) {
    if (this == &orig) {
        return *this;
    }
    delete [] data;
    size = orig.size;
    data = new int [size];
    for (int i = 0; i < size; i++) {
        data[i] = orig.data[i];
    }
    return *this;
}
```

//prevent destroying
data

Call;

```
ECE462XX a; //()
```

```
ECE462XX b = a; //calls copy construct
```



```
b = a; // calls operator =
```

```
b = b; // should have no effect
```

```
virtual ~ECE462XX() { delete [] data; }
```

```
private:
    int size;
    int *data;
};
```

- If using new in constructor, should implement:
- delete destructor
 - Copy constructor
 - operator =

STATIC Attribute
Reference count → NEXT