

C++ "big three":

1. copy constructor
2. destructor
3. assignment operator

like constructor but
gets an object
as argument

MEMORY
MGMT already has memory allocated

constructor:

allocate memory

esp. using new,

if necessary

destructor:

use delete to

free memory

⇒ if no new, prob.
default destr. OK.

assignment op. :

operator = (...) {

① if (this != &rhs) {

standard
alias test

// copy all
elems from
rhs to this

}

②

return *this;

Array class

templated,
just an
array

building our
own version
of STL

vector < >

↑
templates

Typical code separation:

C++ interface:

in the .h

header file

C++ implementation:

in the .cpp file

• h header file:

- is what is seen
by users of the
C++ class ~~or~~ other
programmers

Makefile:

handles: (.CPP)

- compiling!
- linking (.O)
- dependencies

Array class usage:

"driver" program

(main.cpp)

main.cpp:

```
#include <iostream>
```

```
#include "array.h"
```

search
locally

search
/usr/include


```
int main()
```

```
{  
  Array<int> myarr1;  
  Array<float> myarr2;
```



"generic"

"Container"

a lot of these is the
STL

std::cout << myArr;

Array "knows"
how to print itself

overloading
operator <<

```
Array<int> *myarr =  
    new Array<int>;  
    :
```

```
delete myarr;
```

```
} // end of main
```