

UNIVERSITY of WASHINGTON L12: C++ Class Details, Heap CSE333, Summer 2020

Rule of Three

- ❖ If you define any of:
 - 1) Destructor
 - 2) Copy Constructor
 - 3) Assignment (`operator=`)
- ❖ Then you should normally define all three
 - Can explicitly ask for default synthesized versions (C++11):

```
class Point {
public:
    Point() = default;           // the default ctor
    ~Point() = default;         // the default dtor
    Point(const Point& copyme) = default; // the default cctor
    Point& operator=(const Point& rhs) = default; // the default "="
    ...
}
```

5

5

UNIVERSITY of WASHINGTON L12: C++ Class Details, Heap CSE333, Summer 2020

Nonmember Functions

- ❖ “Nonmember functions” are just normal functions that happen to use some class
 - Called like a regular function instead of as a member of a class object instance
 - This gets a little weird when we talk about operators...
 - These do *not* have direct access to the class’ private members
- ❖ Useful nonmember functions often included as part of interface to a class
 - Declaration goes in header file, but *outside* of class definition

10

10

UNIVERSITY of WASHINGTON L12: C++ Class Details, Heap CSE333, Summer 2020

friend Nonmember Functions

- ❖ A class can give a nonmember function (or class) access to its non-public members by declaring it as a **friend** within its definition
 - Not a class member, but has access privileges as if it were
 - friend functions are usually unnecessary if your class includes appropriate “getter” public functions

Complex.h

```
class Complex {
    ...
    friend std::istream& operator>>(std::istream& in, Complex& a);
    ...
}; // class Complex
```

Complex.cc

```
std::istream& operator>>(std::istream& in, Complex& a) {
    ...
}
```

11

11

UNIVERSITY of WASHINGTON L12: C++ Class Details, Heap CSE333, Summer 2020

 Poll Everywhere pollev.com/cse33320su

If we wanted to overload `operator==` to compare two points, what type of function should it be?

- ❖ Reminder that Point has getters and a setter

- non-friend + member
- friend + member
- non-friend + non-member
- friend + non-member
- We’re lost...

12

12

W UNIVERSITY of WASHINGTON L12: C++ Class Details, Heap CSE333, Summer 2020

malloc vs. new

	malloc()	new
What is it?	a function	an operator or keyword
How often used (in C)?	often	never
How often used (in C++)?	rarely	often
Allocated memory for	anything	arrays, structs, objects, primitives
Returns	a <code>void*</code> (should be cast)	appropriate pointer type (doesn't need a cast)
When out of memory	returns <code>NULL</code>	throws an exception
Deallocating	<code>free()</code>	<code>delete</code> or <code>delete[]</code>

25

25

W UNIVERSITY of WASHINGTON L12: C++ Class Details, Heap CSE333, Summer 2020

 Poll Everywhere pollev.com/cse33320su

❖ What will happen when we invoke `bar()`?

- If there is an error, how would you fix it?

A. Bad dereference
B. Bad delete
C. Memory leak
D. "Works" fine
E. We're lost...

```
class Foo{
public:
    Foo::Foo(int val) { Init(val); }
    Foo::~Foo() { delete foo_ptr_; }
    void Foo::Init(int val) {
        foo_ptr_ = new int(val);
    }
    Foo& Foo::operator=(const Foo& rhs) {
        delete foo_ptr_;
        Init(*(rhs.foo_ptr_));
        return *this;
    }
private:
    int* foo_ptr_;
};

void bar() {
    Foo a(10);
    Foo b(20);
    a = a;
}
```

26

26