

Constructors

A constructor (ctor) initializes a newly-instantiated object

A class can have multiple constructors that differ in parameters

Which one is invoked depends on how the object is instantiated

A constructor is always invoked when creating a new instance of an object.

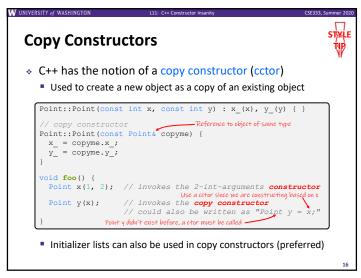
Written with the class name as the method name:

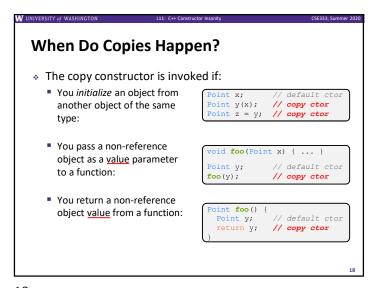
Point (const int x, const int y);

C++ will automatically create a synthesized default constructor if you have no user-defined constructors

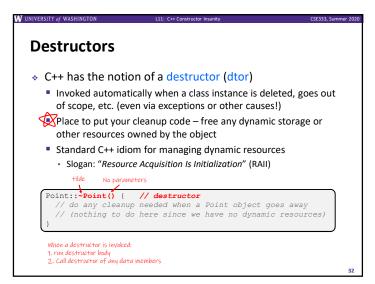
Takes no arguments and calls the default ctor on all non-"plain old data" (non-POD) member variables

Synthesized default ctor will fail if you have non-initialized const or reference data members





18



```
Overloading the "=" Operator
You can choose to define the "=" operator
    But there are some rules you should follow:
      oint& Point::operator=(const Point& rhs) {
       if (this != &rhs) { // (1) always check against this x_ = rhs.x_; Where important when data
                             members are Dynamic memory
        y_ = rhs.y_;
      return *this;
                             // (2) always return *this from op=
                           Should be a reference to *this to allow chaining
                    // default constructor
     Point a;
    a = b = c; // works because = return *this
     a = (b = c); // equiv. to above (= is right-associative)
    (a = b) = c; // "works" because = returns a non-const
                    // reference to *this
    Explicit equivalent:
    a.operator=(b.operator=(c));
                                                                         29
```

29