

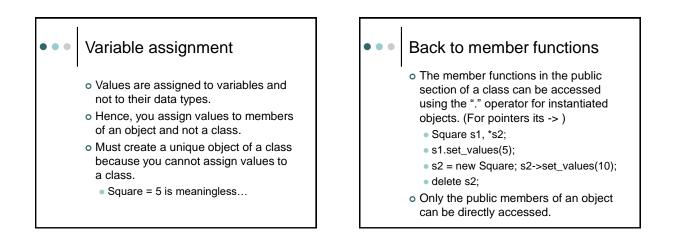
Objects: Reminder

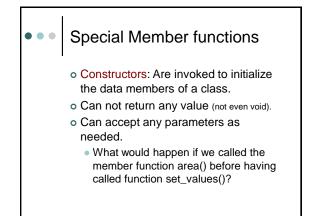
 The data members and member functions of an object have the same properties as the data members and member functions of its class.

Further study: https://www.programiz.com/cpp-programming/object-class

• Assignment operator

- Square s3 = s2;
- By default, copying a class object is equivalent to copying all its elements including pointers.

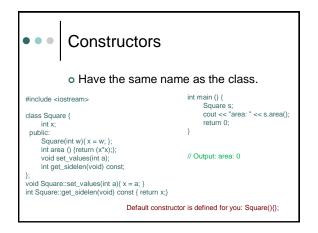


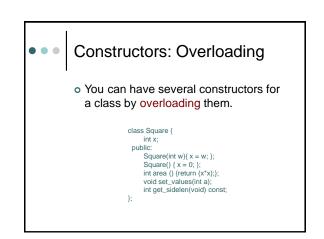


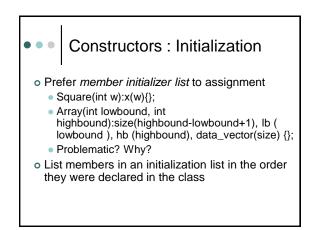
Constructors vs. Destructors

- Constructors initialize the objects in your class.
- Destructors clean up and free memory you may have allocated when the object was created.

Constructors Constructors o Differs from other member functions. • Have the same name as the class. o Initializes a newly created object. #include <iostream> int main () { Square s(3); o Other member functions are invoked class Square { std::cout << "area: " << s.area(); by existing objects. int x; return 0. public: Square(int w){ x = w; }; int area () {return (x*x);}; // Output: area: 9 A Constructor is invoked automatically when an object is created. void set_values(int a); int get_sidelen(void) const; void Square::set_values(int a){ x = a; } int Square::get_sidelen(void) const { return x;}



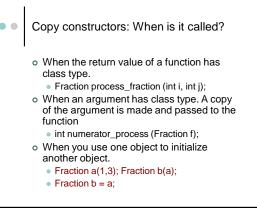


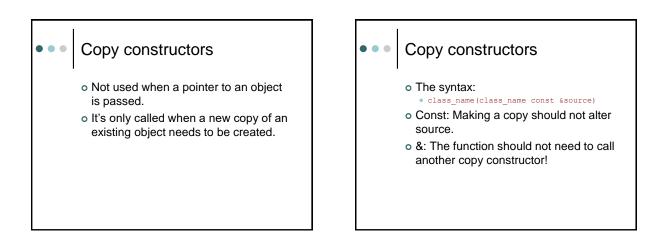


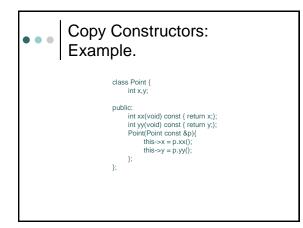
Constructors: Warning If you implement no constructor, the compiler automatically generates a default constructor for you But if you write any constructors at all, the compiler does not supply a default constructor.

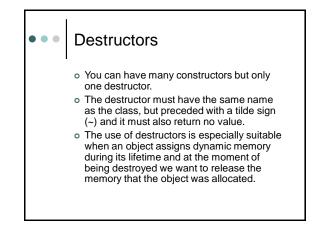
Copy Constructors

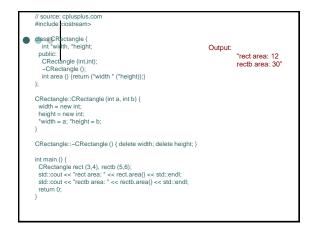
- o Gets called in a number of situations.
- If you do not write one, the compiler automatically generates one for you.





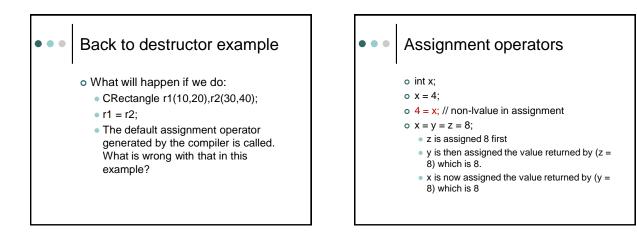


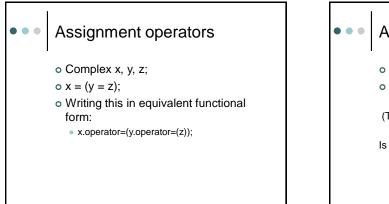




this pointer

- Useful when a member function manipulates two or more objects.
- It holds the address of the object for which the member function is invoked.
- It is always passed to a non-static member function. This ensures the right object is updated using member functions.



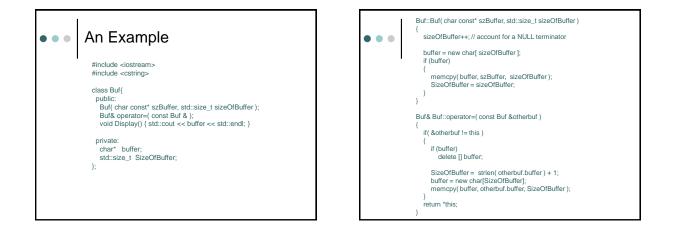


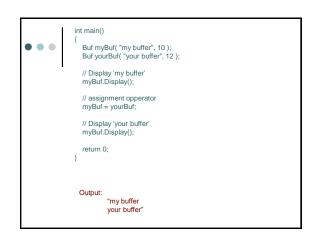
Assignment operators

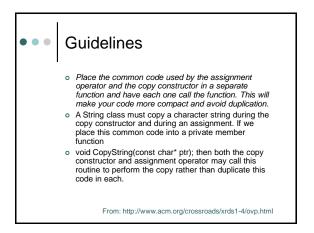
- Square s1 = s2;
- The default behavior : Performs simple member by member copy.
 (This is the one generated by the compiler)

Is the assignment operator the same thing as copy constructor?

Assignment Operator Assignment Operator • • • o Copy constructor initializes an object. o Syntax: class name& operator=(const class name &source) o Assignment operator copies values to o const: Making an assignment should not alter an existing object. source. o &: The function should not need to call a copy • Hence, in some cases: Copy constructor. constructor has to do more work than assignment operator.







Guidelines

- If your class has pointer data, you **must** provide an assignment operator. If writing an assignment operator, you must also write a copy constructor (and destructor?).
- The generated assignment operator performs member-wise assignment on any data members of your class. For pointer variables, we almost always do not want this because the data members of the copy will point to the same data as the copied object!
 Worse, if one of the objects is destroyed, the data is destroyed with it. A run-time error will occur the next time the remaining object tries to access the now nonexistent data.

Guidelines

- When dealing with pointers, Always implement the assignment operator for your class; do not let the compiler generate the default assignment operator. (Remember rule of 3)
- The compiler will generate a default assignment operator for your class if you do not provide one. In order to be in complete control of how your class operates, always provide an assignment operator.

Guidelines

- · Check for assignment to self.
- Disaster can result if a variable is assigned to itself. Consider:
- X x; x = x; Suppose class X contains pointer data members that are dynamically allocated whenever an object is created. Assignment always modifies an existing object. The dynamic data will, therefore, be released before assigning the new value. If we do not check for assignment to self, the above assignment will delete the existing pointer data, and then try to copy the data that was previously deleted!

• Guidelines

- The destructor must release any resources obtained by an object during its lifetime, not just those that were obtained during construction.
- Make the constructor as compact as possible to reduce overhead and to minimize errors during construction.