More on Classes and OOP

10 19 2001

Opening Discussion

- What did we talk about last class?
- The results of the survey indocate that the pace should be slowed down a bit, but that the workload isn't that bad. It will probably be even more reasonable with a slightly slower pace. For this reason I will be cutting chapters 11 and 13 from the schedule (I think).
- Computer programs written in C++.

Anatomy of a Class

- Constructor/Destructor These are special methods that are called when the object is instantiated or when it is destroyed.
- Methods Normal functions for the class. They have special access to the class itself.
- Members Data stored in the class.
- All of these can be public, protected, or private.

Constructor/Destructor

- When an object is created or destroyed special functions are called automatically. These are the constructor and destructor.
- Constructors can take arguments that are used to initialize the object. For example a student might take initial values for the names and grades.
- Destructors typically don't take arguments.

Using a Class

- When we declare a variable with the type of a class we call it an object. We can access the methods and members of an object using the "dot" notation we looked at yesterday.
- When we call a method the execution goes into it like a normal function. Remember that inside the function though we don't have to use a "dot" to access the methods and members of that object.

A Copy for Every Object

- One thing to keep in mind with methods is that each object has its own copy of every member (we'll discuss an exception to this on the next slide).
- This should make intuitive sense when thinking about objects. For example each student object needs its own name and grade, but it is good to keep in mind that what appears to be the same variable in the same function call will have different values depending on what object the method is invoked for.

static in Classes

- If you want only one copy of a variable for an entire class of objects you would use the static keyword.
- The static keyword in a class is much like it is in a function, it specifies that only one copy of that variable should exist in the program.
- It can also be used for methods. A static method can be invokes without an object, but can only change static variables.

```
Student::totalStudents
Student::getHighestGrade()
```

Overloading Functions

- C++ allows for functions to be overloaded. This means that you can use the same function name for two different functions as long as they have different *signatures*. That means that the compiler can figure out which one to use based on the parameters passed in. That means two functions must either have different numbers of arguments or different types.
- You can do this with any function but it should be used with caution. It is required for constructors where the *name* is fixed.

Minute Essay

- What did we talk about today? Over the weekend think about the things you do in an average week and how you could express them as algorithms. Also think about what information you would need in a class to instantiate an object that was you.
- What time between 2pm and 6pm on Monday works best for you for a review?