Operator Overloading

10 29 2001

Test Results

- Tests results were generally quite good with well over half of the two classes scoring in the As or Bs.
- Median of 83.

Opening Discussion

- What did we talk about last class (before the test)?
- Do you have any questions about assignment #4?
- Right now I’m going to try to stick to the topics currently listed on the web page (the ones I posted last week). Notice the days off and let me know what you think about this.
Wouldn't it be nice

- When you write a class, it would occasionally be nice to be able to use objects of that class with normal operators so that we can write more normal looking expressions.
- For example if you have a class for complex numbers it would be nice to write \( a-b+c \) where \( a, b, \) and \( c \) are objects of type complex. In C++ you can.

Operators as Functions

- In C++, operators are treated just like other functions, and just like other functions you have the ability to overload them so that they do different things with different arguments.
- The naming of operators is fairly straightforward.

  ```
  int operator+ (int a, int b);
  int operator== (Student a, Student b);
  ```

Operators as Methods

- Operators can be declared as methods of a class. In this case a binary operator takes only one argument which is the second operand.
- The downfall of this is that you can't overload an operator where the first argument isn't of the class in question. For that you have to do something a bit different.
Friends

- It is possible to give functions outside of a class, or even other classes the ability to see private data in a class. This is done using the friend keyword.
- Using friends is generally considered a bad thing and should only be done as a last resort. Like non constant globals, having too many friends in your design indicates that there are problems.

Operators as Friend Functions

- One valid use of friends can be to create overloaded operators that don't have an object of that class as the first argument.

```cpp
class Example {
   friend bool operator==(int j, Example &e);
   private:
      int a;
};
bool operator==(int j, Example &e) {
   return j==e.a;
}
```

Minute Essay

- What did we talk about today?
- Geoffrey Marcy will be sneaking on campus twice today about searching for extrasolar planets (4:00pm in the Science Lecture Hall and 7:30pm in Laurie Auditorium). I will give 5 points on the test for students showing up to one of those talks and 7 points if you show up to both. You will have to find me at the talks.