



THE UNIVERSITY OF VERMONT
COLLEGE OF ENGINEERING &
MATHEMATICAL SCIENCES

Operator Overloading

CS 124 / Department of Computer Science

Motivation

We create objects and would like certain operators to "understand" how to work with these objects.

For example, in our introduction to C++ we overloaded two operators to work with our lecturer class:

- Stream insertion operator, <<
- Less than operator, <

Why can't we use these without overriding?

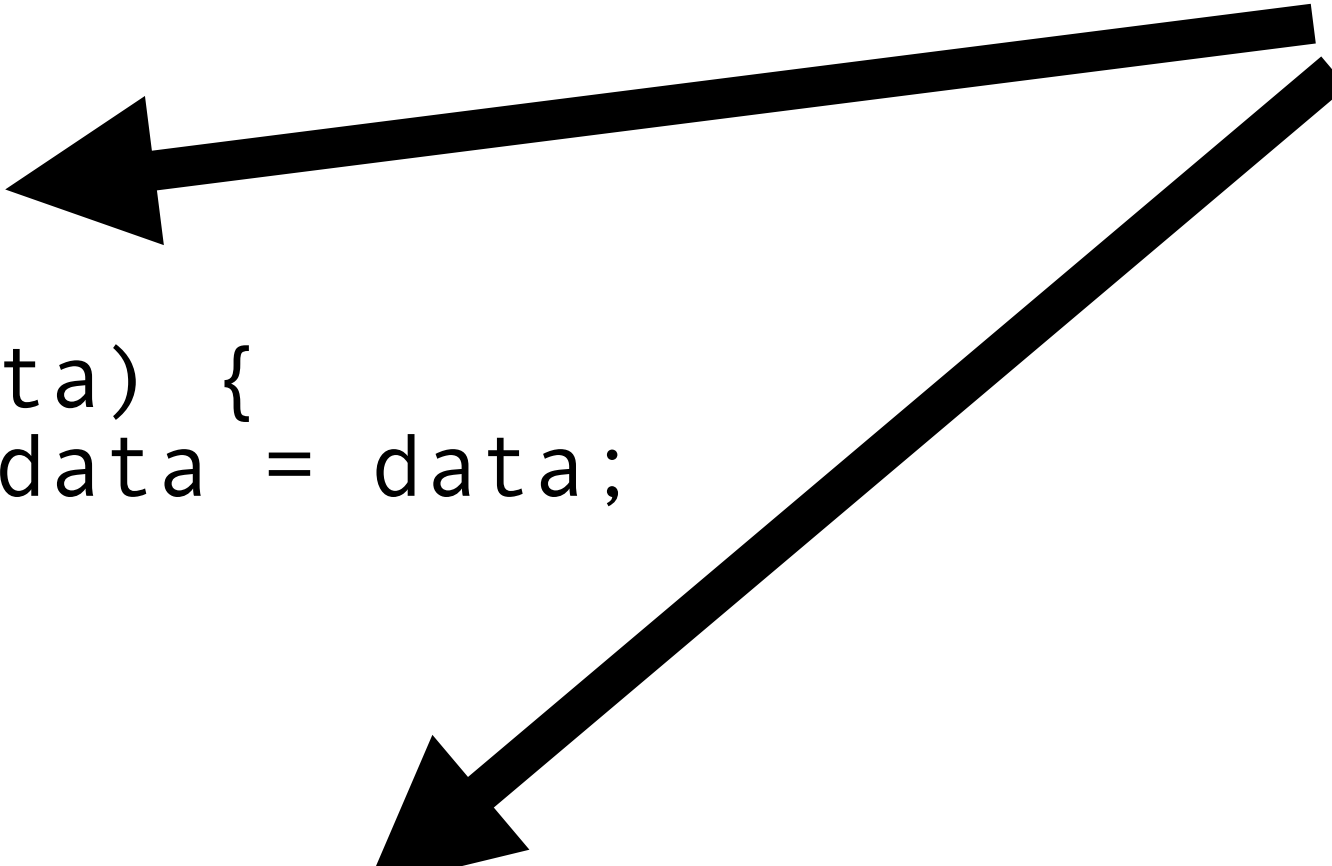
Motivation

```
#include <iostream>

class Foo {
private:
    int data;
public:
    Foo(int data) {
        this->data = data;
    }
};

int main() {
    Foo f1 = Foo(42);
    Foo f2 = Foo(77);
    assert (f1 < f2);
    return 0;
}
```

There seems to be a natural ordering of Foo objects based on the numeric values of their data fields



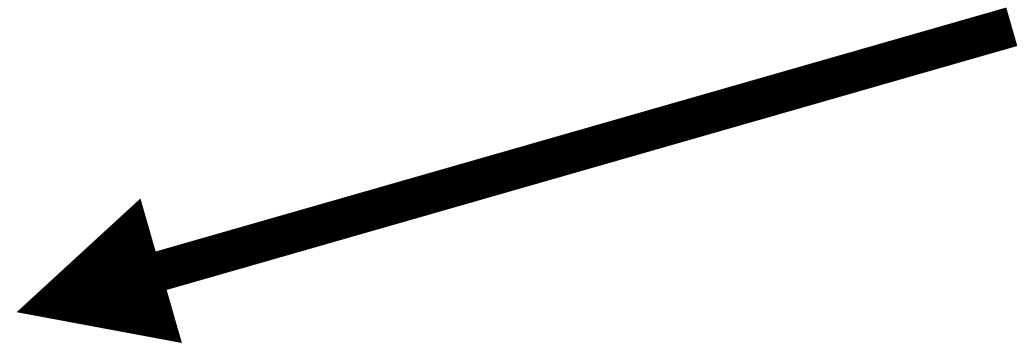
Motivation

```
#include <iostream>

class Foo {
private:
    int data;
public:
    Foo(int data) {
        this->data = data;
    }
};

int main() {
    Foo f1 = Foo(42);
    Foo f2 = Foo(77);
    assert (f1 < f2);
    return 0;
}
```

So it would make sense if we could compare Foos in this way. Is $f1 > f2$? Is $f1 < f2$? Does $f1 == f2$? and so on.



Motivation

```
#include <iostream>

class Foo {
private:
    int data;
public:
    Foo(int data) {
        this->data = data;
    }
};

int main() {
    Foo f1 = Foo(42);
    Foo f2 = Foo(77);
    assert (f1 < f2);
    return 0;
}
```

But this raises an error! "Invalid operands to binary expression ('Foo' and 'Foo')."

We can't compare our Foos!

How do we fix this?



How do we fix this?

Operator



Overloading

Operator Overloading

By overloading an operator, we show C++ how to use it in the context of our class.

For example, now we can override the < operator so we can compare Foos.

```
friend bool operator < (const Foo& lhs, const Foo& rhs) {  
    return lhs.data < rhs.data;  
}
```


Operator Overloading

```
friend bool operator < (const Foo& lhs, const Foo& rhs) {  
    return lhs.data < rhs.data;  
}
```

```
friend bool operator > (const Foo& lhs, const Foo& rhs) {  
    return lhs.data > rhs.data;  
}
```

Operator Overloading

```
friend bool operator <= (const Foo& lhs, const Foo& rhs) {  
    return lhs.data <= rhs.data;  
}
```

```
friend bool operator >= (const Foo& lhs, const Foo& rhs) {  
    return lhs.data >= rhs.data;  
}
```

Operator Overloading

```
friend bool operator == (const Foo& lhs, const Foo& rhs) {  
    return lhs.data == rhs.data;  
}
```

```
friend bool operator != (const Foo& lhs, const Foo& rhs) {  
    return lhs.data != rhs.data;  
}
```