

CS368 Lecture 10

Wednesday 5 November 2014

Reminders:

- P3 due tonight
- P4 to be assigned later today, due in 2 weeks

Last class:

- More overloading operators
- Makefiles

Today:

- Overloading ==
- “Friends”
- Input/output streams
- Console Input/Output
- Overloading output <<

Overloading Review

Member vs. Non-member

What is really happening?

Overloading == and other relational operators

Use:

```
if (p1 == p2) { ... }
```

In header file:

```
friend bool operator==(const Polynomial& lhs,  
                        const Polynomial& rhs);
```

In source file:

```
bool operator==(const Polynomial& lhs,  
                const Polynomial& rhs) {  
  
    if (lhs.size != rhs.size) { return false; }  
  
    for (int i=0; i<lhs.size; i++) {  
        if (lhs.coefs[i] != rhs.coefs[i])  
            return false;  
    }  
  
    return true;  
}
```

Why friend?

The `friend` keyword

- Not a member method!
- Yet has access to private/protected members of a class
- Not called using dot or `→` (unless ...)
- Declared in class granting access
- Not affected by visibility keywords
- Not symmetric

Console Input and Output

Streams

cin and cout

Extraction (>>), Insertion (<<) operators

Overloading <<

Use:

```
cout << p1 << " is a polynomial" << endl;
```

In the header file:

```
// in the public section, member (accessor)
void print (ostream& out = cout) const;

// non-member function (outside class definition)
ostream& operator<< (ostream& out, const Polynomial& p);

// or you can just use friend
```

In the source file:

```
ostream& operator<< (ostream& out, const Polynomial& p) {
    p.print(out);
    return out; // why?
}

void Polynomial::print(ostream& out) const {

    if (size == 0) return;

    for (int i=size-1; i > 0; i--)
        out << coefs[i] << "x^" << i << " + ";
    out << coefs[0] << endl;
}
```

Remember, `cout` is a predefined instance of the `ostream` class.

Understanding `cin`

```
char c;  
int x;  
double y;  
  
cin >> c >> x >> y;
```