Object Oriented Programming in C++ 2

Seonghyun Park

dd December, 2018

Outline

1. Operator Overloading
   - Motivation
   - Operator overloading in C++
   - Syntax

Today's Goal

- Learn about operator overloading

Motivation

- Do you like this code?

```
struct Point {
  double x, y;

  // implementation elided
  Point& add(const Point& rhs);
  Point& add(const Point& rhs);
};

p.add(q).sub(r);
// Maybe we can write as (p + q) - r?
```

Motivation

- How about this?

```
struct List {
  // implementation elided
  int& elemAt(size_t n) const;
  const int& elemAt(size_t n) const;
};

lst.ememAt(3);
// Maybe we can write as lst[3]?
```

Operator Overloading

- Operators in c++?
  - Binary operators +, −, ∗, /, |, &, ||, &&, >, <, ==, ...
  - Unary operators −, , !, ++, −−, ...
  - Assignment operators +=, −=, ∗=, /=, ...
  - Array subscriptor operator `[]`
  - Call operator `()`
- Don't be surprised. There are 38 operators (until C++20) that you can overload.
- See `https://en.cppreference.com/w/cpp/language/operators`

Operator Overloading Syntax

- Overloaded operator can be implemented in forms of either member functions or non-member functions.
  - Non-member functions: `operator+`, `operator<<` (for I/O), etc.
  - Memmber functions: `operator[]`, `operator+=`, `operator()`, etc.
- Has naming convention `operatorXX` for an operator `XX`.

Operator overloading Syntax (Cont'd)

- Each operator has its desired function types. Please follow the rules unless you have very very good reasons.
- The best way to learn operator overloading is learning-by-doing. Now, let's get code.