Lecture-5

- Operator overloading
- Standard template library
  - string
  - vector
  - list
  - iterators

Operator overloading

- On two objects of the same class, can we perform typical operations like
  - Assignment (=), increment (++), decrement(--)
  - Write to a stream (<<)
  - Reading to a stream (>>)
- Can be defined for user defined classes.

  ⇒ Operator overloading
- Most of the common operators can be overloaded.
- Operators – can be member/non-member functions

Operator overloading … cont.

- Arity of operator
  - Number of parameters required.
- Unary operators – take one argument
  - *E.g.*, ++, --, !, ~, etc.
  - C unary operators remain unary in C++
- Binary operators – take two arguments.
  - *E.g.*, =, >, <, +, -, etc.
  - C binary operators remain binary.
- Typical overloaded operators
  - +, -, >, <, +=, ==, !=, <=, >=, <<, >>, [ ]

Operator functions rules

- Member function operators
  - Leftmost operand must be an object (or reference to an object) of the class.
  - If left operand is of a different type, operator function must NOT be a member function
- Built-in operators with built-in data types CANNOT be changed.
- Non-member operator function must be a **`friend`** if
  - **`private`** or **`protected`** members of that class are accessed directly

Syntax

- Member function

```
return_type classname :: operator  symbol (args)
  {
    // code
  }
```

- Non-member function

```
return_type operator symbol (args)
  {
    // code
  }
```

Example

```
class Integer
   {
        private:
           int value;
        public:
           Integer (int val) : value (val) {  }
           void operator ++( ) { value++; }  // Member op
           friend Integer operator +       // Non-member op
                (const Integer& i, const Integer& j);
   };

   Integer operator + (const Integer&i, const Integer& j)
   {
          return Integer (i.value + j.value);
   }
```

Standard template library

- Defines many useful classes.
- Popular among them
  - string, vector, list, map, iterators, etc.
  - Each of these is a class.
  - Has many useful functions.
- Reference:

http://www.processdoc.com/doc/cppstl/index.html

It lists all the functions, coding examples and many nice features for strings, vectors, lists and iterators.