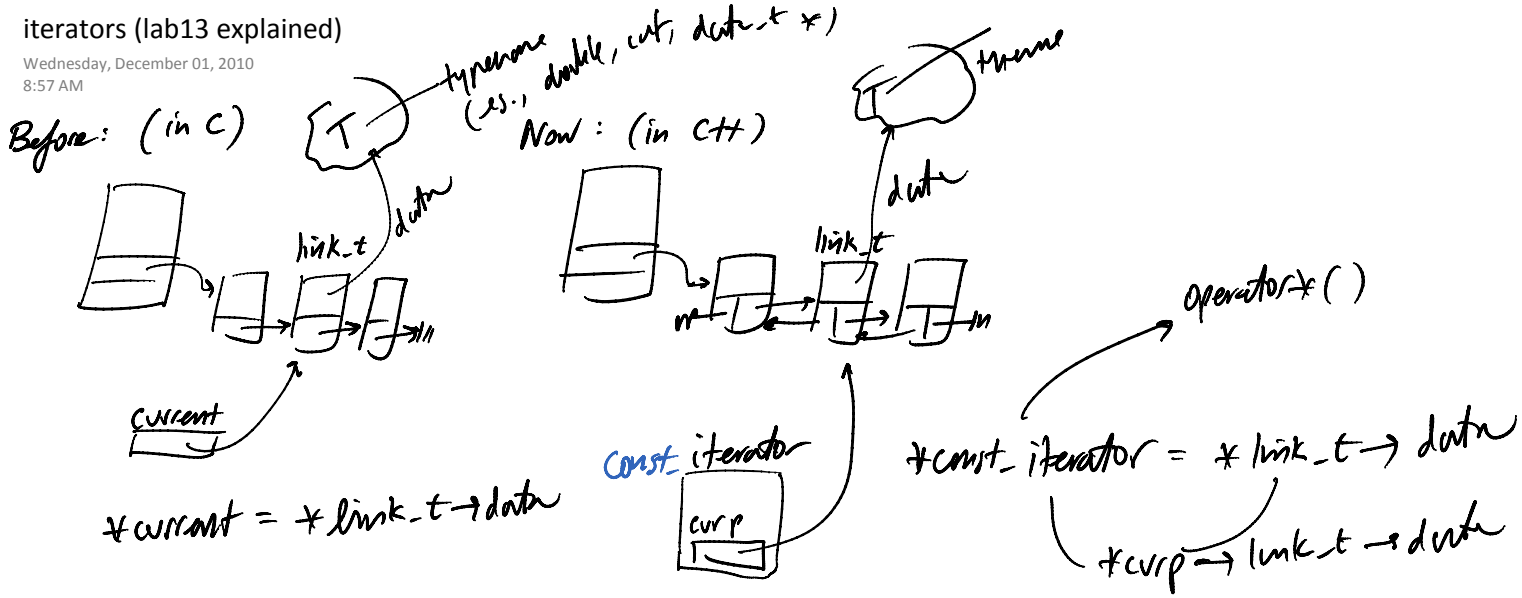
iterators (lab13 explained)

Wednesday, December 01, 2010
8:57 AM



- iterator is just a wrapper for a link_t * pointer
- can add extra logic to typical ptr operations to make iterator behave like a "smart pointer"
- iterator is just another C++ object or class

class list_t

```
:
public:
class const_iterator
{
  public:
  const_iterator(): curr(NULL) { };
  const_iterator(link_t *p): curr(p) { };
  //operators
  const T& operator*() const
  { return retrieve(); }
 (*) protected:
  private:
  link_t *curr;
  T& retrieve() const
```

overloaded constructors

got moved to protected: section so iterator: public const_iterator could inherit

```
T& retrieve() const
{ return curp->data; }
friend class list_t<T>;

};

class iterator : public const_iterator
{
  ...
};

const_iterator& operator++()          // ++itr
{ curp = curp->next; return *this; }

const_iterator operator++(int)        // itr++
{ const_iterator old(*this); ++(*this); return old; }
```

no ass