

# CSC 601 Fall 2002 Midterm #1 Solutions

#1.

```
struct IIteratorR
// Read-only iterator.
{
    virtual T get() const throw (runtime_error) =0 ;
    virtual bool atEnd() const =0 ;
    virtual void advance() throw (runtime_error) =0 ;
};

struct IIteratorRW : public IIteratorR
// Read-write iterator.
{
    virtual T& get() throw (runtime_error) =0 ;
};

struct ILockSet
{
    // virtual dtor
    virtual ~ILockSet() {} ;

    // inspectors
    virtual int size() const throw() =0 ;
    virtual IIteratorR* new_permanent_iterator() const throw() =0 ;
    virtual IIteratorRW* new_transient_iterator() const throw() =0 ;

    // mutators
    virtual void addPermanent( T item ) throw (runtime_error)=0 ;
    virtual void addTransient( T item ) throw (runtime_error)=0 ;
    virtual bool remove( T item ) throw (runtime_error) =0 ;
};

struct ILSFactory
{
    virtual ILockSet* new_LockSet() const =0 ;
};

class ListLSIterator : public IIteratorRW
// NOTE: Accessing this as IIteratorR* hides the mutating get().
{
public:
    ListLSIterator( const ListLockSet& lls ) {}

    virtual T get() const throw (runtime_error) { /*...*/ }
    virtual T& get() throw (runtime_error) { /*...*/ }
    virtual bool atEnd() const throw() { /*...*/ }
    virtual void advance() throw (runtime_error) { /*...*/ }
};

class ListLockSet : public ILockSet
{
public:
    ListLockSet() { /*...*/ }
    ~ListLockSet() { /*...*/ }

    // inspectors
    int size() const throw() { /*...*/ }
    virtual IIteratorR* new_permanent_iterator() const throw() { /*...*/ }
    virtual IIteratorRW* new_transient_iterator() const throw() { /*...*/ }

    // mutators
    void addPermanent( T item ) throw (runtime_error) { /*...*/ }
    void addTransient( T item ) throw (runtime_error) { /*...*/ }
    bool remove( T item ) throw (runtime_error) { /*...*/ }

private:
    std::set<T> _transients, _permanents ; // or whatever. . .
};

struct ListLSFactory : public ILSFactory
{
    ListLockSet* new_LockSet() const { /*...*/ }
};
```

#2.

```
(define cartesian-product
(lambda (flist)
  (lambda (xlist)
    (if (null? xlist)
        null
        (cons ((car flist) (car xlist))
              ((cartesian-product) (cdr flist)) (cdr xlist))))))
```

#3.

```
sctor stwist 50
ictor itwist 5 50
sctor stwist 70
ictor itwist 7 70
itwist 7 70
sdctor 0
```