**Aggregation vs Composition**

With/without pointers in C++

**Aggregation**

To qualify as an **aggregation**, a whole object and its parts must have the following relationship:

* The part (member) is part of the object (class)
* The part (member) can belong to more than one object (class) at a time
* The part (member) does *not* have its existence managed by the object (class)
* The part (member) does not know about the existence of the object (class)

**Without a pointer**

## Aggregation

class Member;

class Chief

{

public:

Chief(Member& ch) : obj(ch) {}

private:

Member& obj;

};

## Composition

#include "obj.h"

class Chief

{

private:

Member obj;

};

**With a pointer**

Aggregation

class Member;

class Chief

{

public:

Chief(Member\* ch) : obj(ch) {}

Chief() : obj(NULL) {}

void setMember(Member\* ch) { obj = ch; }

private:

Member\* obj;

};

## Composition

class Member;

class Chief

{

public:

Chief();

~Chief();

void renewMember();

private:

Member\* obj;

};

// --- source file

#include "obj.h"

Chief::Chief()

: obj(new Member)

{

}

Chief::~Chief()

{

delete obj;

}

void Chief::renewMember()

{

delete obj;

obj = new Member;

}