

public:

float checkbalance();

Account

```
graph TD; Account((Account)) --- checking((checking)); Account --- savings((savings));
```

checking

savings

~~Data~~



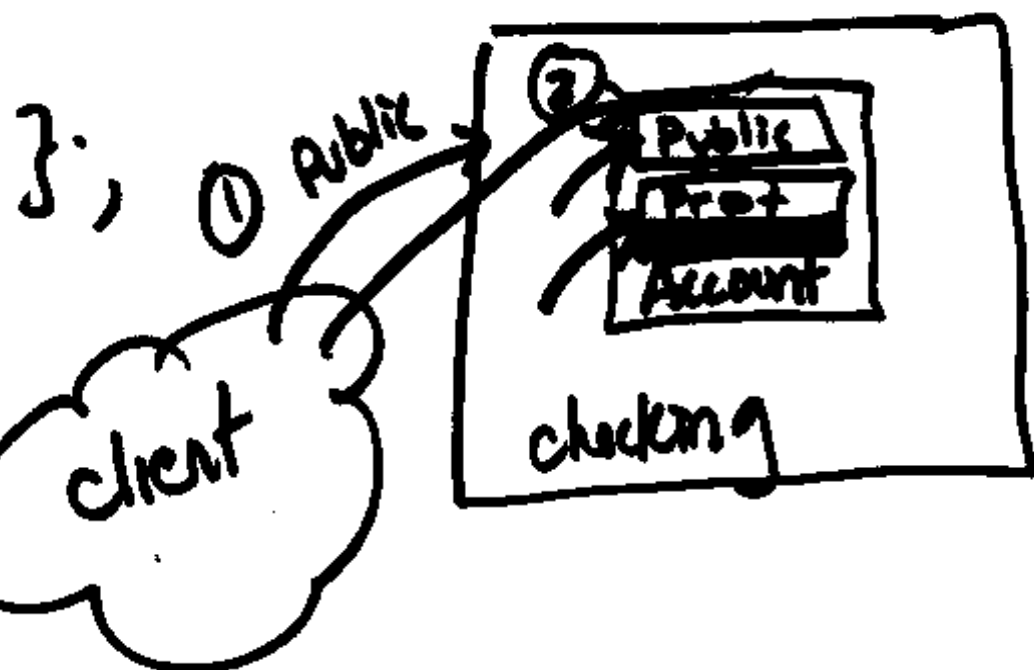
checking obj;
obj.checkbalance();

Derivation List

class checking : public account
{

Public :
protected :
private :

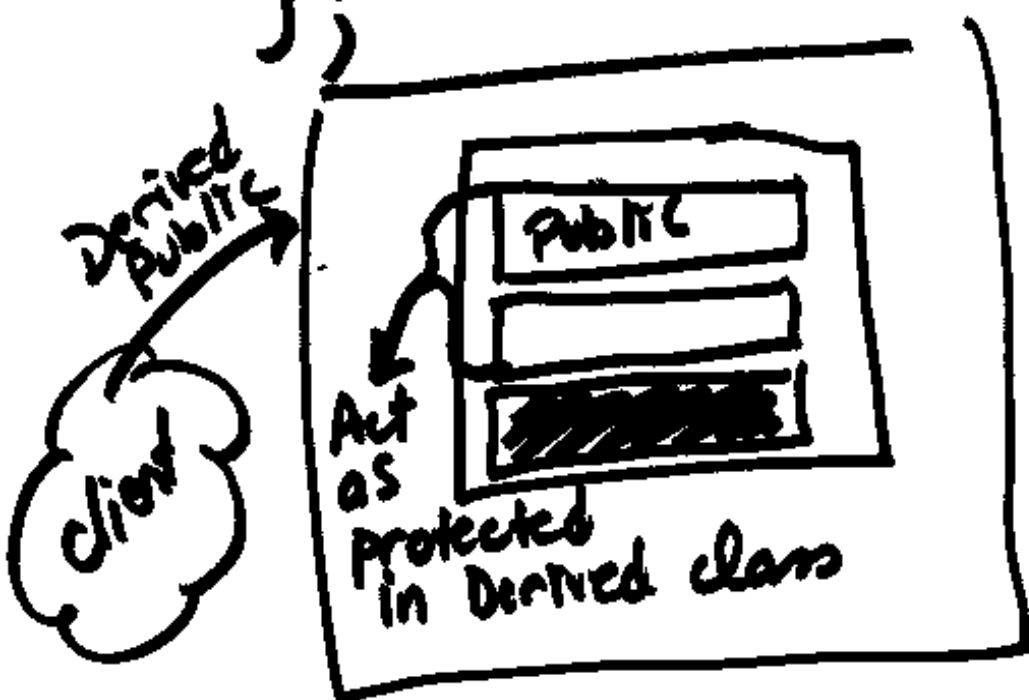
Public
Derivation



class checking : protected account
{

protected
derivation

};



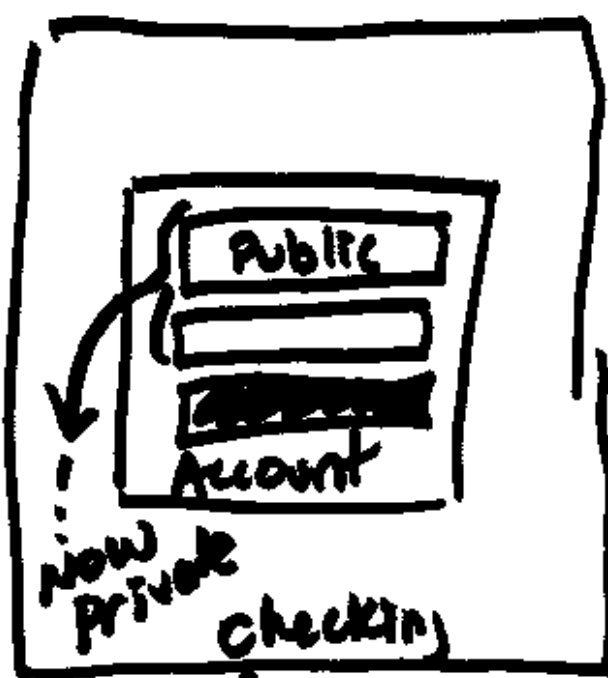
class checking : private account

{

↑
Private
derivation

};

Public
checking



client

Derived

checking
Public is protected

Call

checking obj;



function(obj);

function (account &)

function (account & arg)

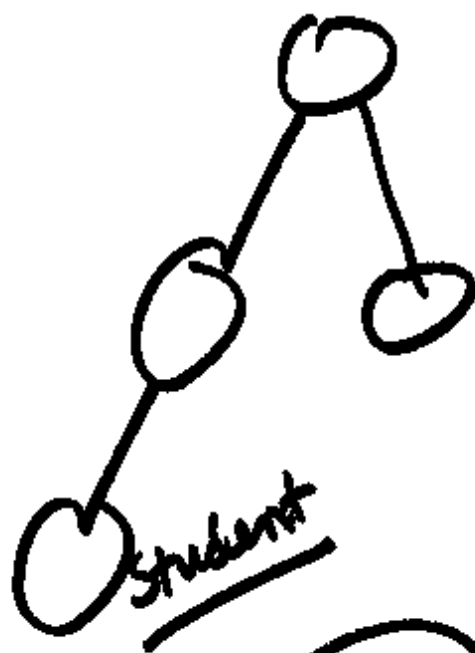
{
arg = ~~func~~ public func();
Account class
public function

Account

arg. checking func();
syntax error!

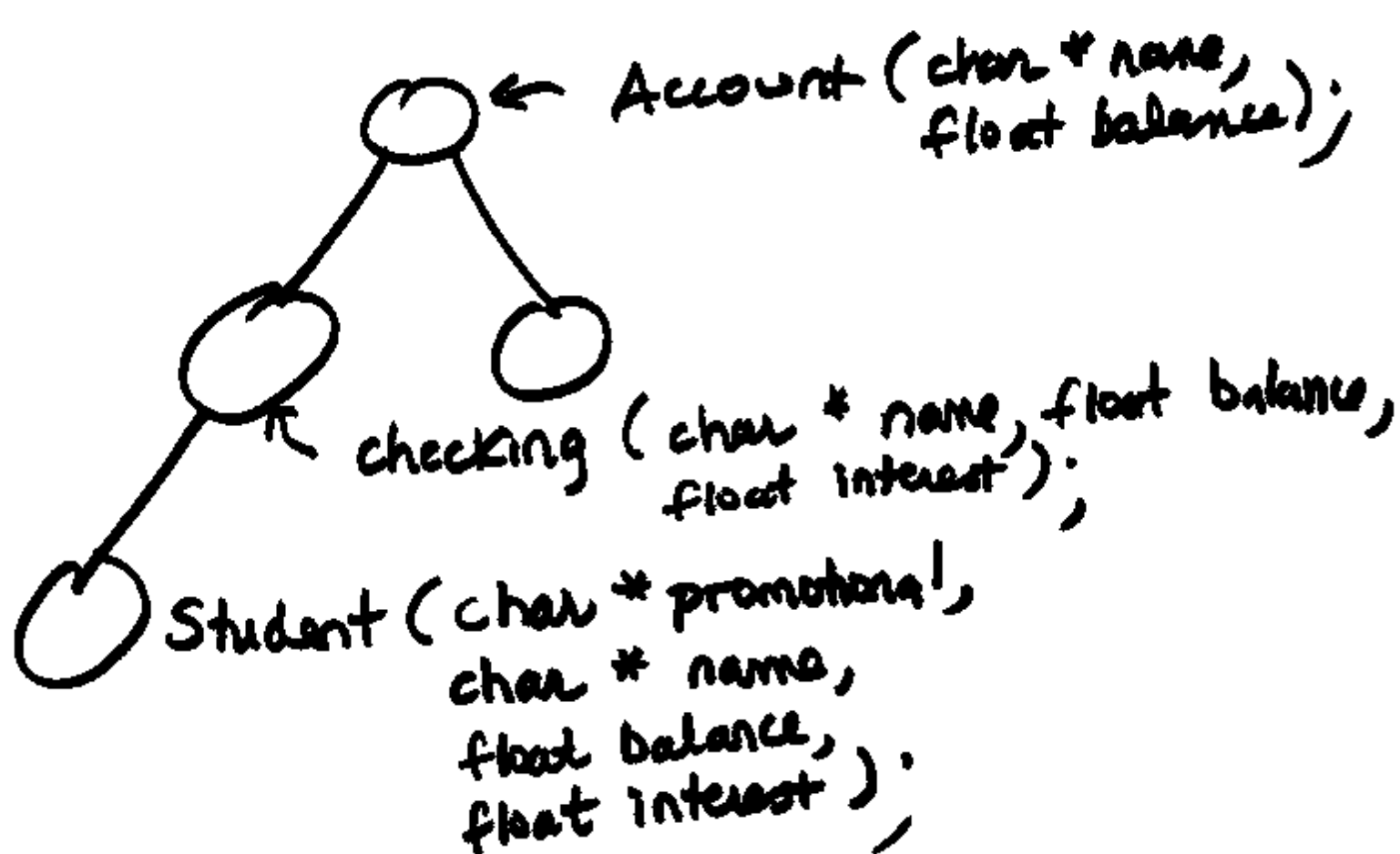
Default Constructors

OR



Account() ✓
checking() ✓
then
Student() ✓

Student obj;



① what if.....

Student obj ("DISC1", "Sam Smith", 100.0, 0.0)



~~NO~~ vs. Student obj;

if you have NO initialization Lists



Correct - Initialization List

checking.cpp (Not in .h)

```
checking::checking(char *name, float balance, float intr)
{
    Account(name, balance),
    datamember_interest(intr),
}
}
root = NULL;
```

↑
root(φ)

• h file for checking

Student.cpp

```
student::student(char *pro, char *name,
float bal, float intr) : checking(name, bal,
intr)
{
}
}
```

File



Implement the
Member functions

g++ *.cpp

Class list

{

public:

struct node

{

};

class

Nesting

private

Tree

root → head = new node;
NO!

};

root → head = list::node();

#include <iostream>

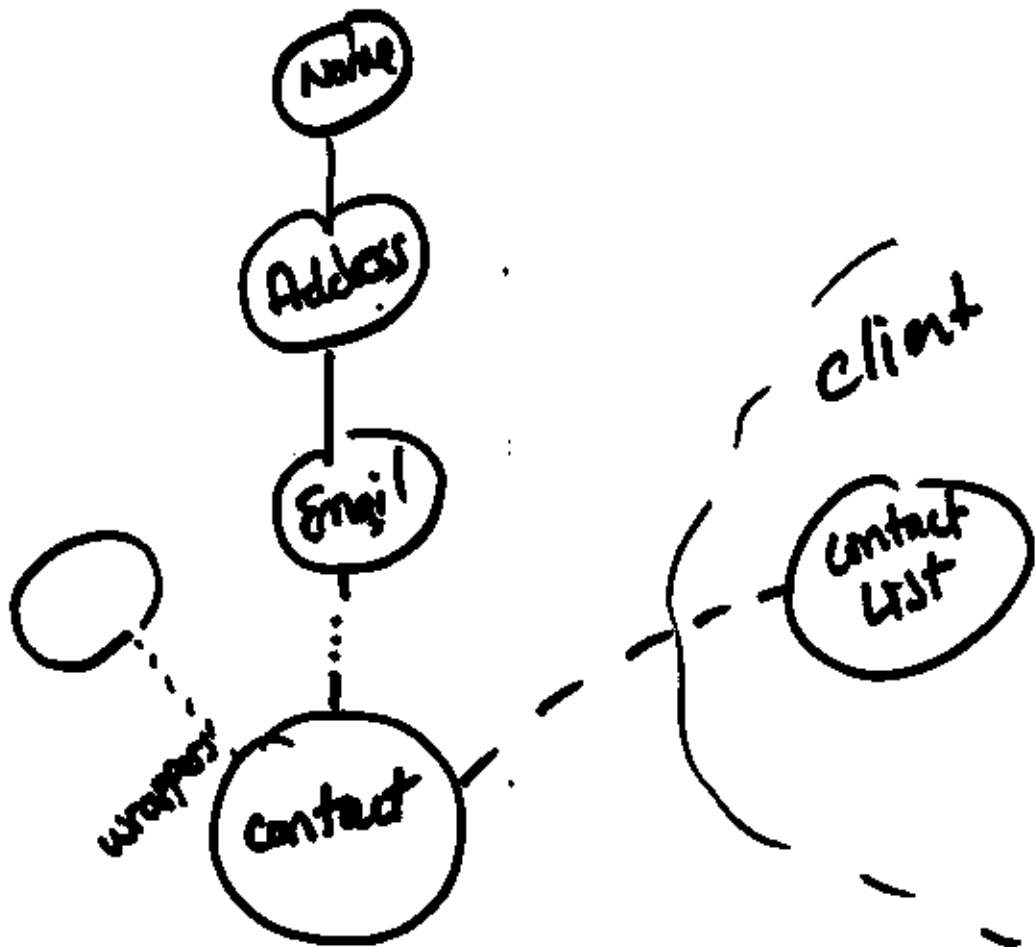
using namespace std;

cout << "...";

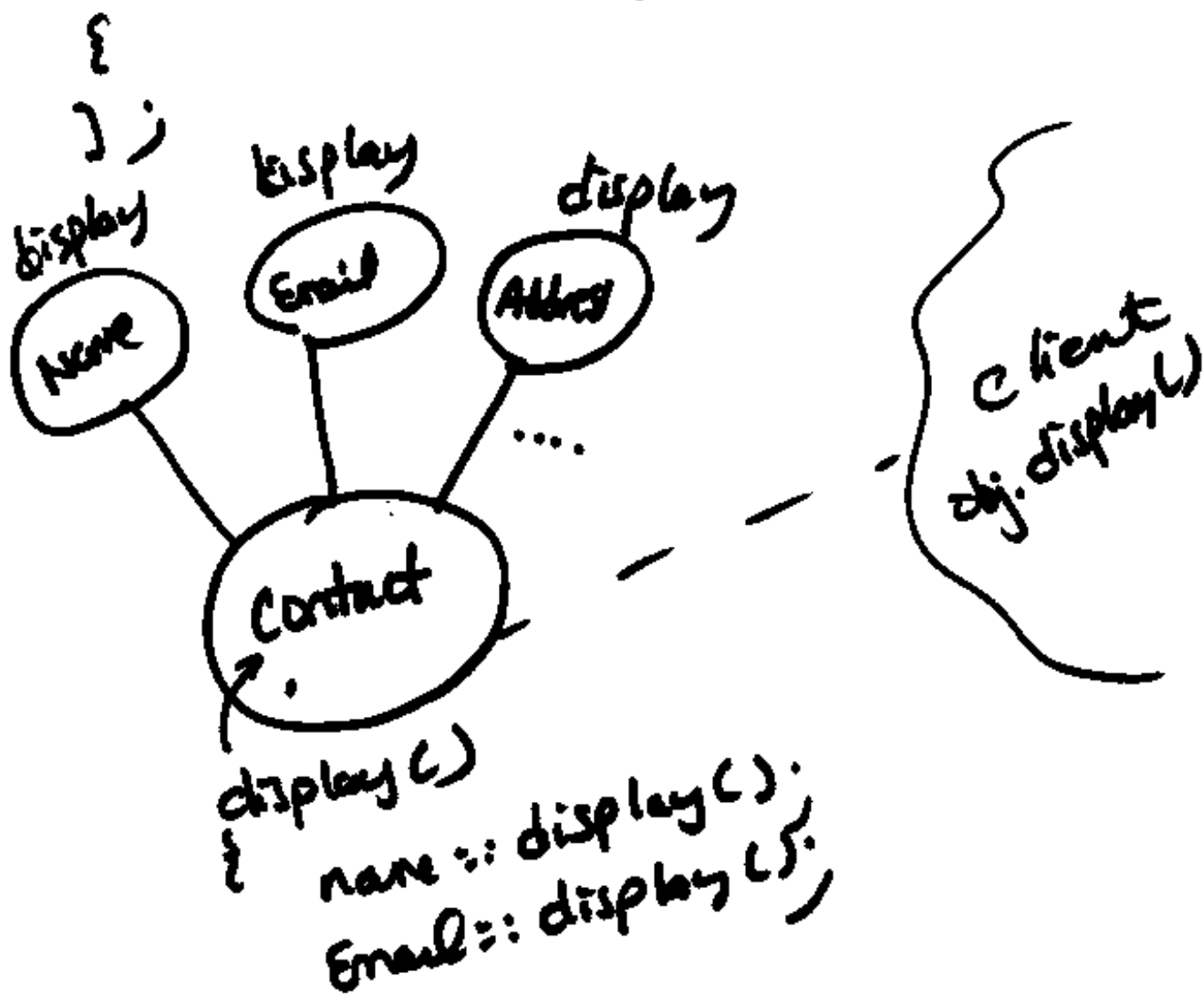
std::cout << "..."

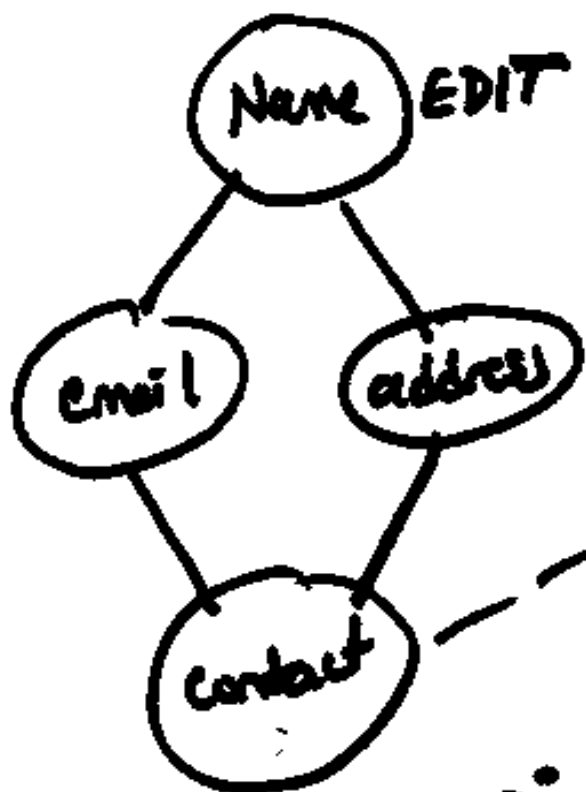


#include <iostream.h>

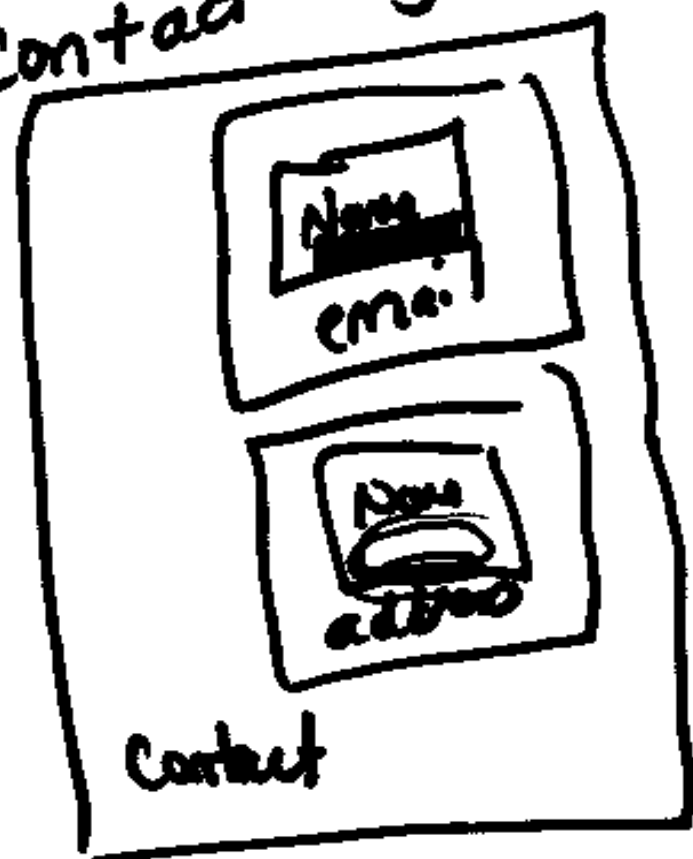


class contact : public name, public email,
public address



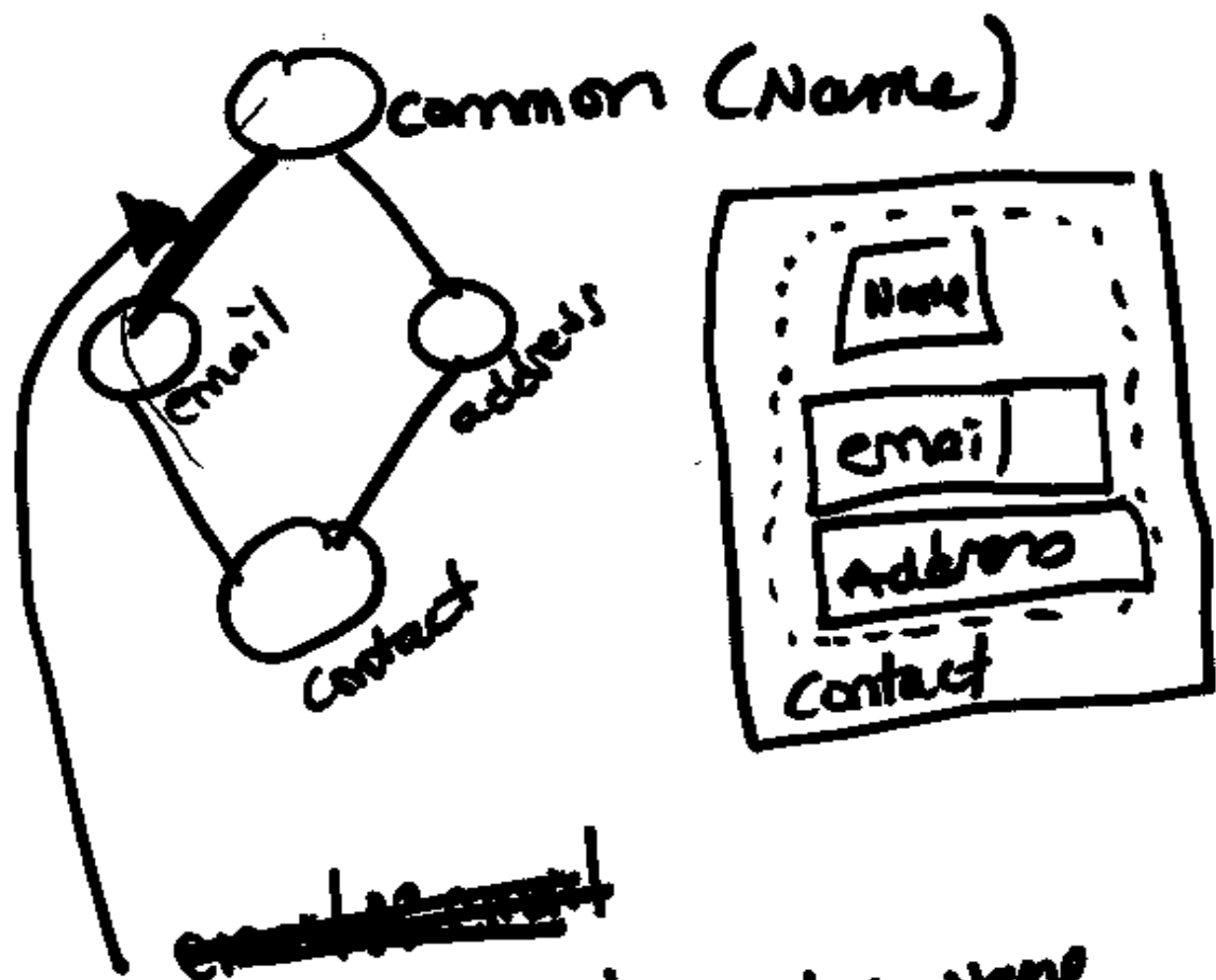


Contact obj



Virtual Inheritance

- common base class



~~example~~

```
class email : virtual public Name
{
}
class address : virtual public Name
{
}
```

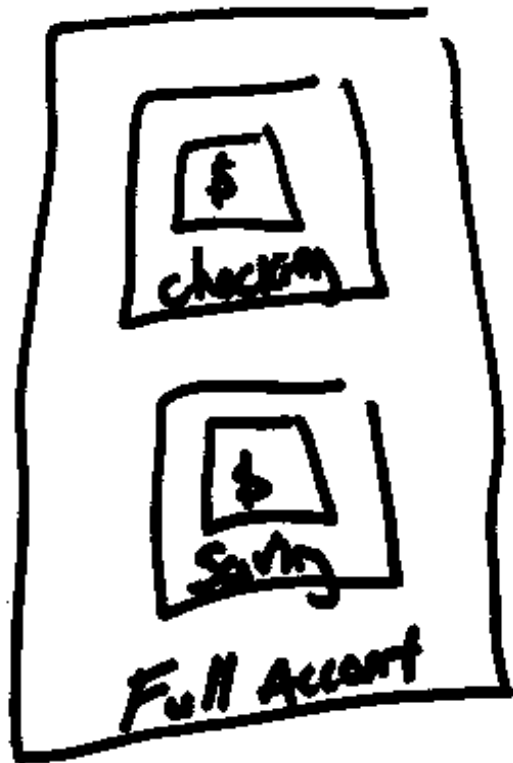
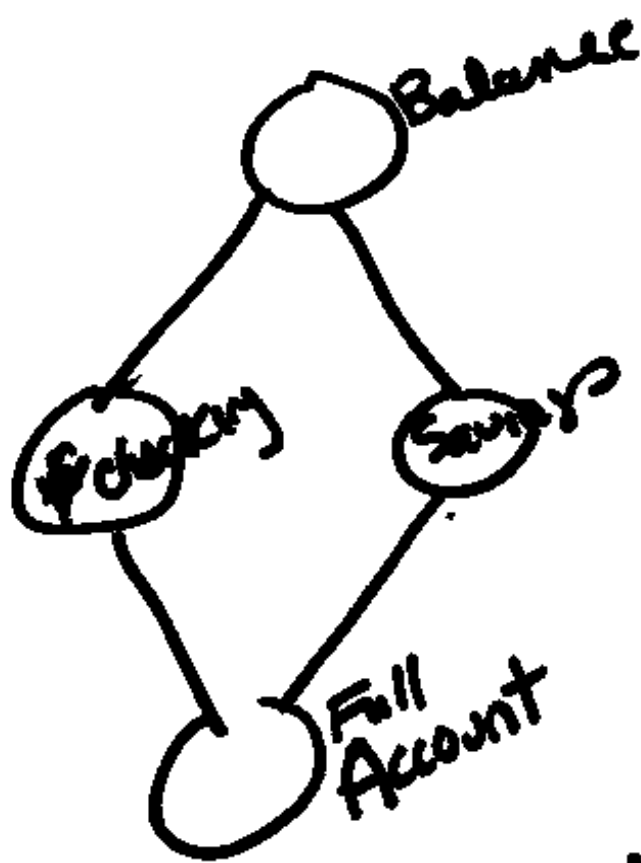
↑ virtual inheritance

class Contact : public email, public address
{
=
=
};

contact obj;

name

Contact **MUST** have initialization
Lists for the common ANCESTOR



HAS A
is Better !!!

