POLYMORPHISM

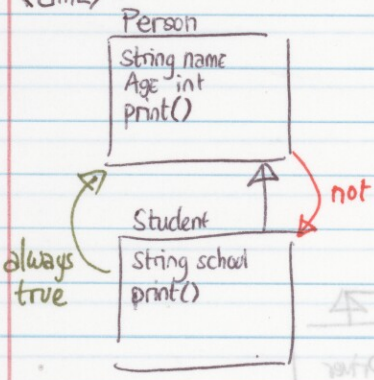
```
Person p1 = new Person()
p1.print();          ; - behaves like person
p1 = new Student()
p1.print();          ; - behaves like student
```

✓ every student is a person

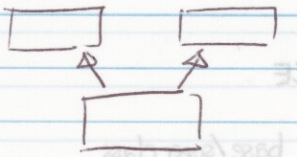
```
Student s2 = new Student();
// s2 = new Person();   ✗ → problem : person may not be student
```

8/30/2010

<UML>



Multiple inheritance



Yes in C++

Not in Java

class/object
encapsulation
inheritance
polymorphism

reuse

C++

```
#include "person.h"
#include <iostream>

Person::Person(string ln, string fn)
  : lastName(ln), firstName(fn)
{
}

Person::~Person()
{

}

void Person::print() {
  cout << "last name: " << lastName << endl;
}
```

all function except constructor should be virtual

destructor

[invoked when delete is called]

```
#ifndef PERSON_H_
#define PERSON_H_
#include <string>

class Person {
  public:
    Person(string ln, string fn);
    virtual ~Person();
    virtual void print();
  protected:
    const string lastName;
    const string firstName;
};
```

supports polymorphism

!!

#endif /*PERSON_H_*/ ?