



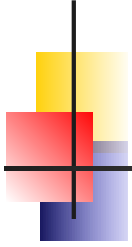
UML Class Diagram by Example



A Single Class

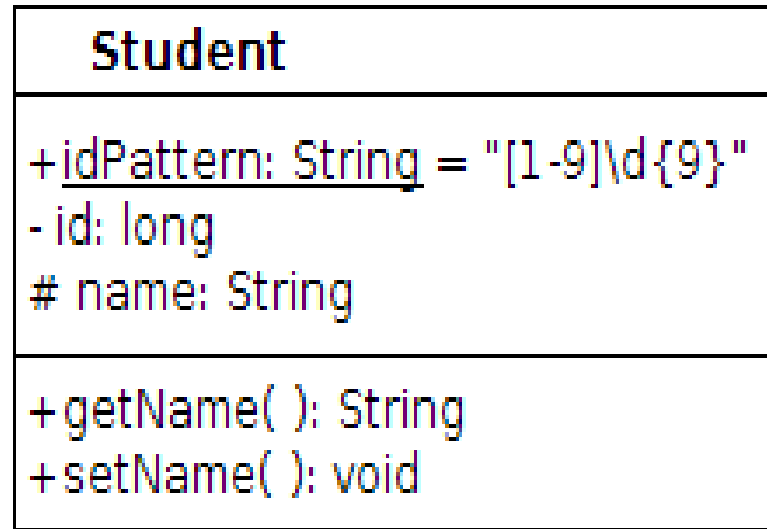
Draw a UML class diagram of this class.

```
public class Student {  
    public static String idPattern = "[1-9]\\d{9}";  
    private long id;  
    protected String name;  
  
    public String getName( ) { . . . }  
  
    public void setName(String aname) { . . . }
```



A Single Class

Draw a UML class diagram of this class.

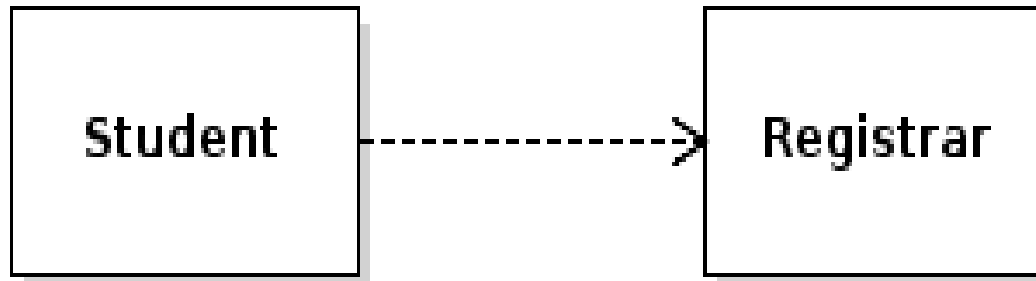




Class with Dependency

A Student uses the Registrar to get his Courses, but he doesn't save a reference to it.

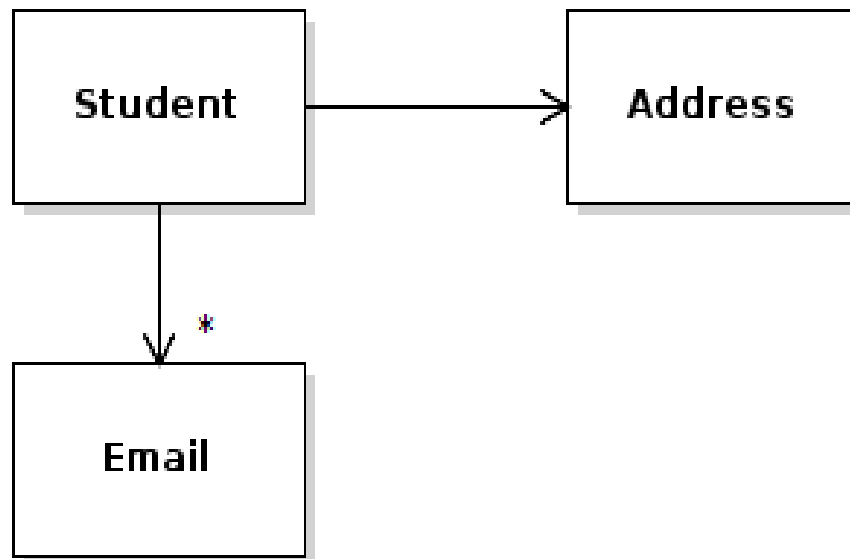
```
public class Student {  
    private long id;  
    //NO Registrar attribute!  
  
    public void addCourse(Course course) {  
        Registrar regis = Registrar.getInstance();  
        regis.enroll(this, course);  
    }  
}
```



Class with Association

A Student *has* an Address and 0 or more Emails.

```
public class Student {  
    private Address homeAddress;  
    /** his email addresses. He may have many. */  
    private List<Email> emails;  
}
```



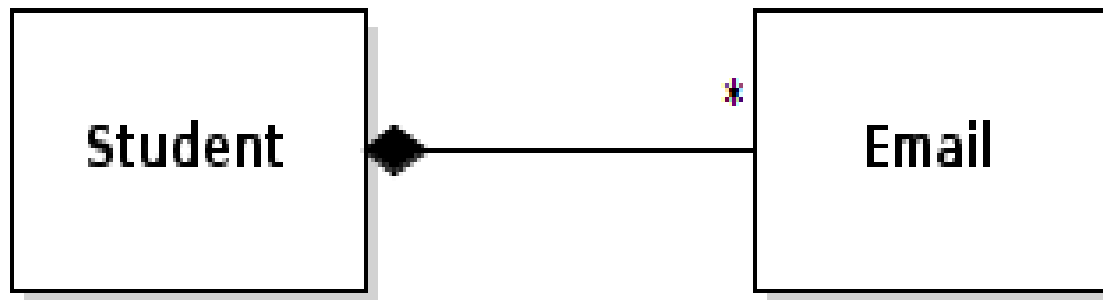
A Student **owns** his Email Addresses

Composition: A Student **owns** his Email addresses and when he is deleted we delete his addresses, too!

```
public class Student {  
    /** student uniquely owns his email addresses*/  
    private List<Email> emails;
```

Modeling:

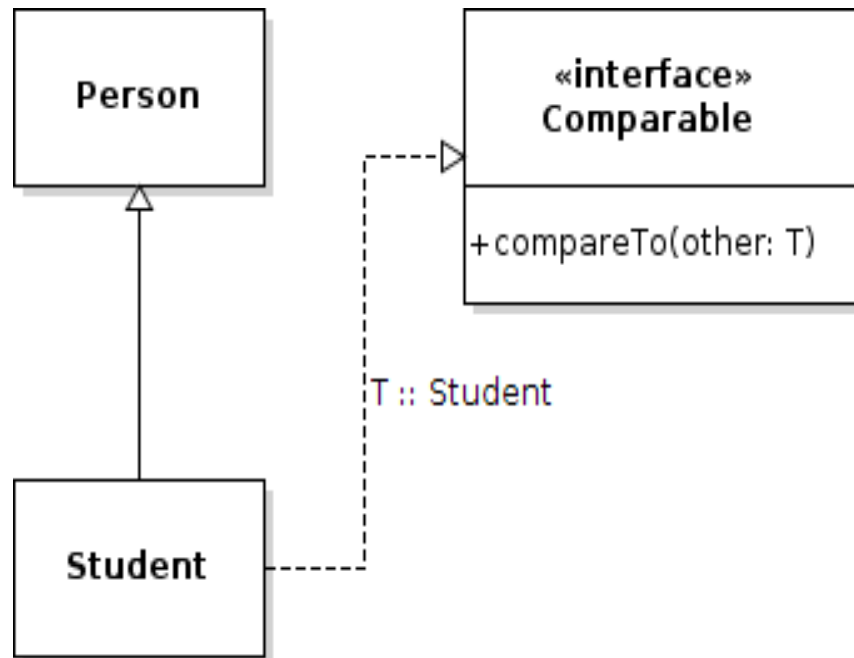
Composition shows "*ownership*" or "*is composed of*" (e.g.: a game board is composed of squares). Be *careful* about using it.



Inheritance & Implements

Student is a subclass of Person

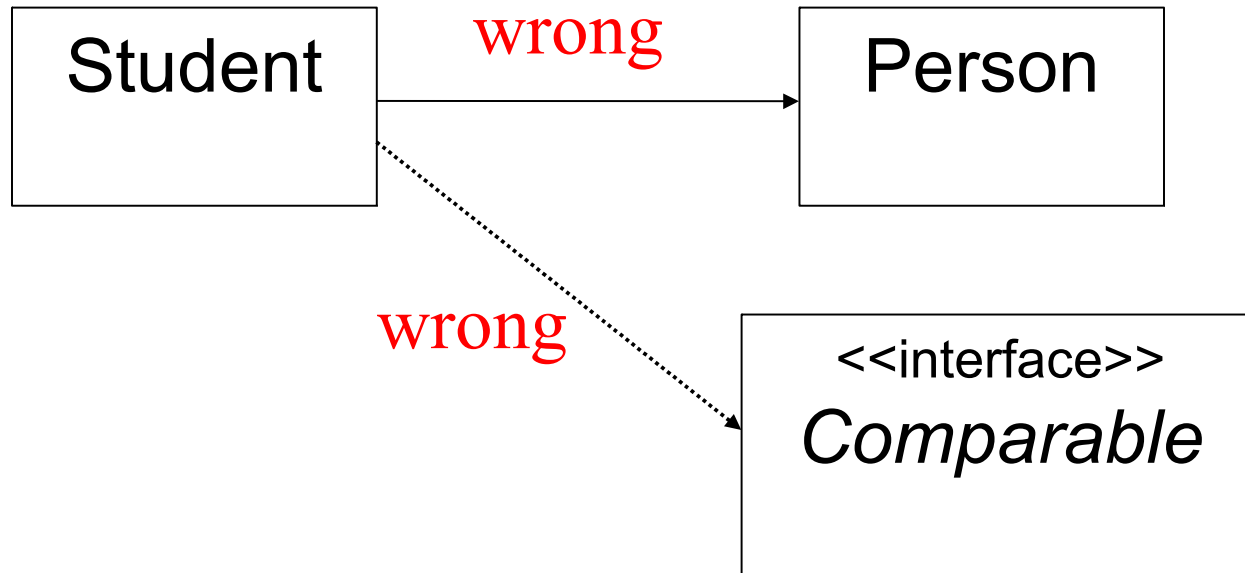
```
public class Student extends Person  
    implements Comparable<Student> {
```



Errors

A UML diagram is for communication.

To communicate clearly, use the **correct notation**.



No partial credit for wrong relationships or bad notation.



Reference

UML Distilled, 3rd Edition. Chapter 3 & 5 cover UML class diagrams.