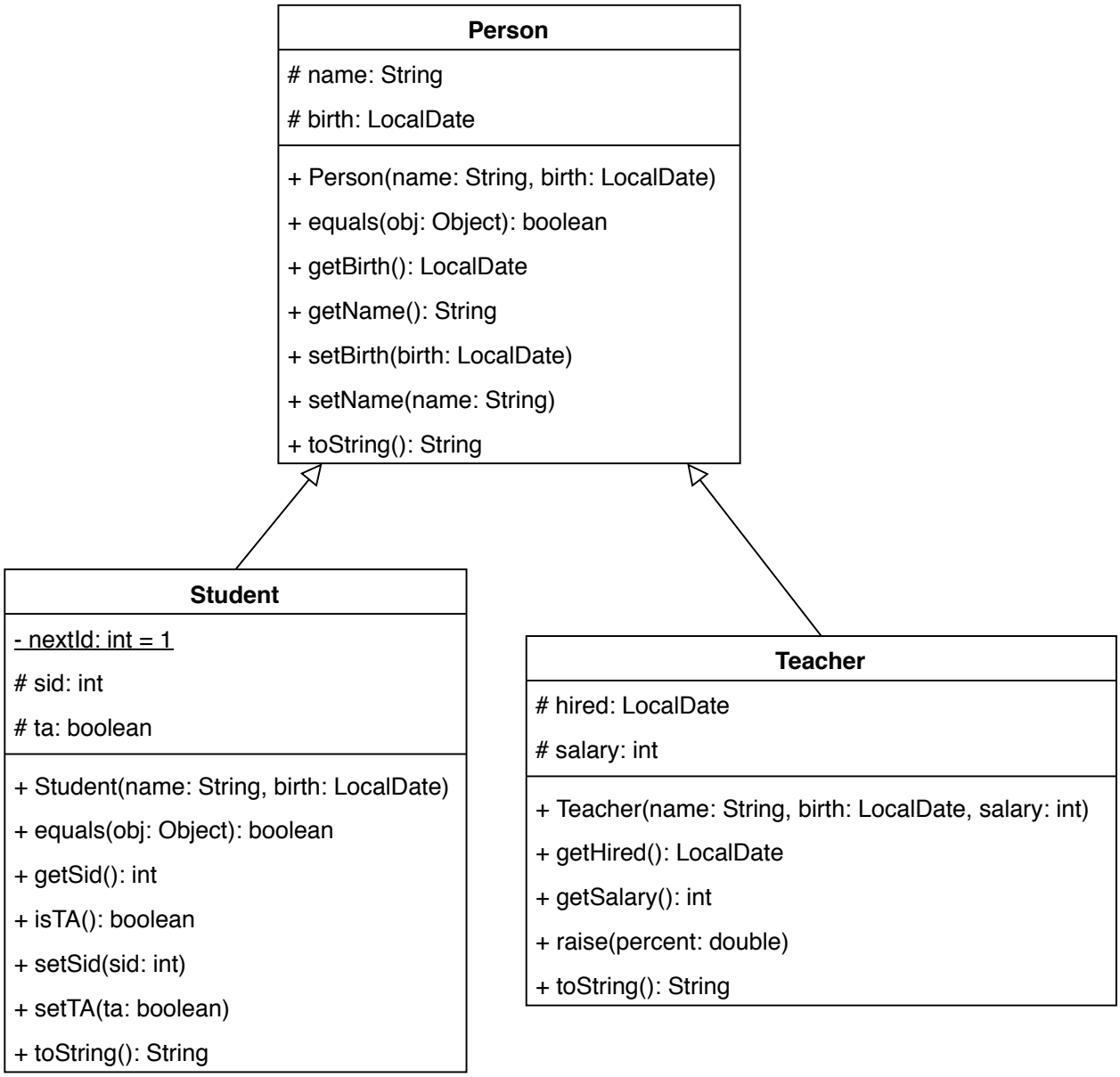
Model 1 Students and Teachers



Questions (15 min) Start time:

1. Based on the UML diagram:

a) What attributes does a `Student` object have? name, birth, sid, ta

b) What attributes does a `Teacher` object have? name, birth, hired, salary

c) Which methods does `Student` override? equals and toString

d) Which methods does `Teacher` override? toString

2. Based on the UML diagram:

a) Which methods does a `Student` and a `Teacher` have in common? (i.e., inherited)

getBirth, getName, setBirth, setName

b) Which methods does a `Student` object have that a `Teacher` object does not have?

getSid, isTA, setSid, setTA

c) Which methods does a `Teacher` object have that a `Student` object does not have?

getHired, getSalary, raise

3. Fill in each blank with either "is a" or "has a":

a) `Person` has a `String`

b) `Person` has a `LocalDate`

c) `Student` is a `Person`

d) `Student` has a `String`

e) `Teacher` is a `Person`

f) `Teacher` has a `LocalDate`

4. Explain the difference between "is a" and "has a" in the previous question.

"Is a" refers to inheritance; every subclass is an instance of its parent class. "Has a" refers to composition; an object's attributes may reference other objects.

5. Why would it be incorrect to say "`Person` is a `Student`"?

Inheritance relationships are one-way. Every `Student` is a `Person`, but that doesn't make every `Person` a `Student`.

6. Which `equals` method (in which class) will be invoked by the following code? Explain your reasoning based on the applicable "is a" or "has a" relationship.

```
LocalDate d = LocalDate.parse("1949-01-17");
Teacher t1 = new Teacher("Anita Borg", d, 123456);
Teacher t2 = new Teacher("Anita Borg", d, 123456);
System.out.println(t1.equals(t2));
```

It will invoke the `equals` method in the `Person` class. There is no `equals` method in the `Teacher` class, but `Teacher` is a `Person`. So `Teacher` inherits the `equals` method of `Person`.