1 Polymorphism: Intuition

```
1 Student s = new Student("Stella");
2 ResidentStudent rs = new ResidentStudent("Rachael");
3 rs.setPremiumRate(1.25);
4 s = rs; /* Is this valid? */
5 rs = s; /* Is this valid? */
```

Which one of *s* = *rs* (**Line 4**) and *rs* = *s* (**Line 5**) is *valid*? And which one is *invalid*?

Hints:

- **Line 1**: What *kind* of address can *s* store? [`Student`]

  ∴ The context object *s* is *expected* to be used as:

  * `s.register(cse114)` and `s.getTuition()`

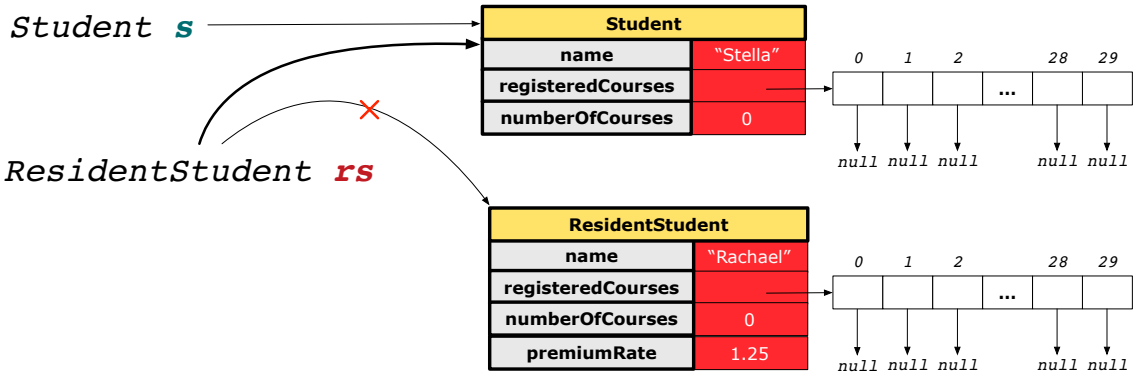
- **Line 2**: What *kind* of address can *rs* store? [`ResidentStudent`]

  ∴ The context object *rs* is *expected* to be used as:

  * `rs.register(cse114)` and `rs.getTuition()`
  * `rs.setPremiumRate(1.50)` [increase the premium rate]

1.1 Problematic Scenario when Allowing *rs* = *s* (Line 5)

- ***rs*** = ***s*** (**Line 5**) should be *invalid*:



- Since ***rs*** is declared as `ResidentStudent`,

  a subsequent call ***rs***`.setPremiumRate(1.50)` can be expected.

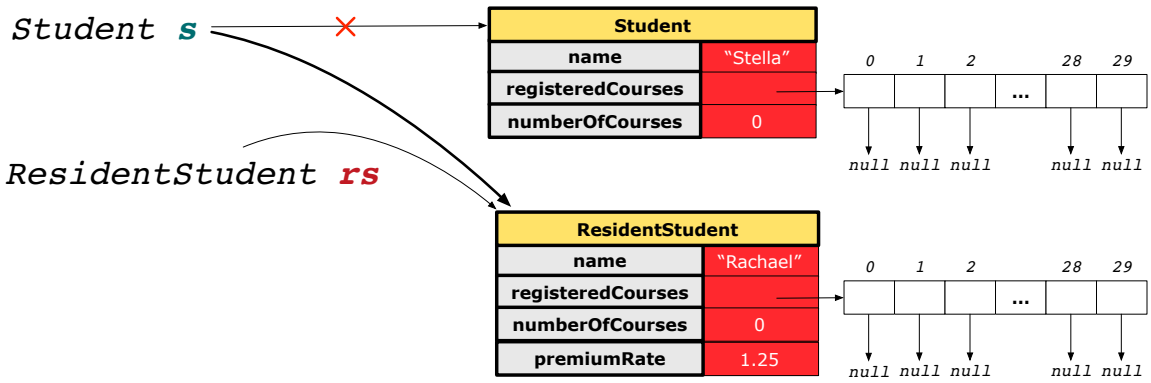
- ***rs*** is now pointing to a `Student` object.

- Then, what would happen to ***rs***`.setPremiumRate(1.50)`?

  CRASH ∵ ***rs***`.premiumRate` is *undefined*!!

1.2 Acceptable Scenario when Allowing *s* = *rs* (Line 4)

- ***s*** = ***rs*** (**L4**) should be *valid*:



- Since ***s*** is declared as `Student`,

  a subsequent call ***s***`.setPremiumRate(1.50)` is ***never*** expected.

- ***s*** is now pointing to a `ResidentStudent` object.

- Then, what would happen to ***s***`.getTuition()`?

  OK ∵ ***s***`.premiumRate` is just *never used*!!

2 Dynamic Binding: Intuition

```
1 Course cse114 = new Course("CSE114", 100.0);
2 Student s;
3 ResidentStudent rs = new ResidentStudent("Rachael");
4 NonResidentStudent nrs = new NonResidentStudent("Nancy");
5 rs.setPremiumRate(1.25); rs.register(cse114);
6 nrs.setDiscountRate(0.75); nrs.register(cse114);
7 s = rs; System.out.println(s.getTuition()); /* output: 125.0 */
8 s = nrs; System.out.println(s.getTuition()); /* output: 75.0 */
```

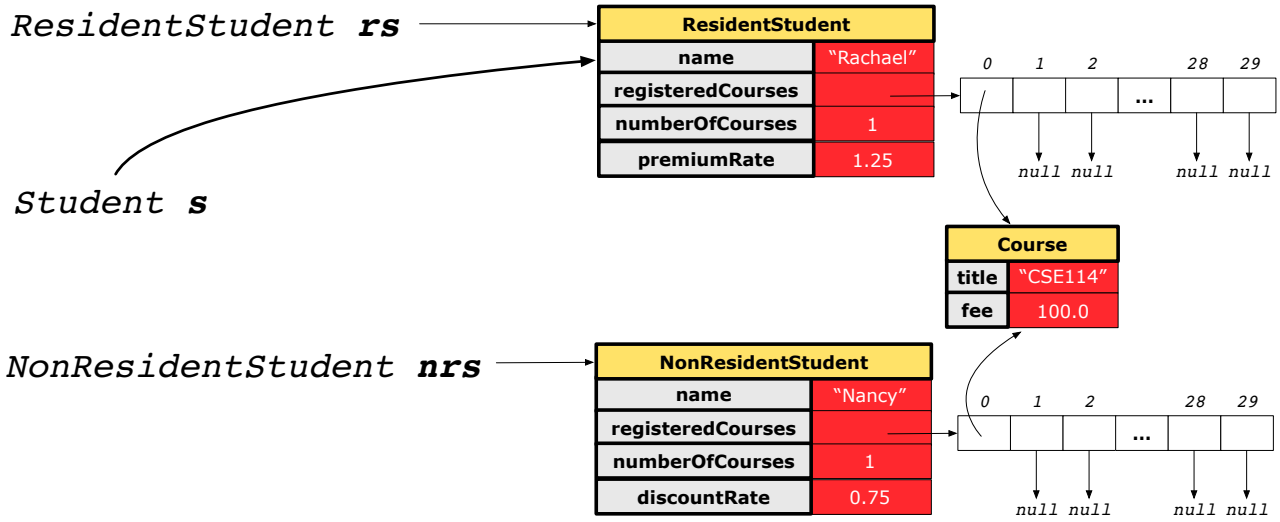
The return values of method call `s.getTuition()` in **Line 7** and **Line 8** depend on where the reference variable `s` is **actually pointing to**

(i.e., a `ResidentStudent` object or a `NonResidentStudent` object).

2.1 After executing `s = rs` (Line 7)

After `s = rs` (**Line 7**), `s` points to a `ResidentStudent` object.

⇒ Calling `s.getTuition()` applies the `premiumRate`: 100.0×1.25.



2.2 After executing `s = nrs` (Line 8)

After `s = nrs` (**Line 8**), `s` points to a `NonResidentStudent` object.

⇒ Calling `s.getTuition()` applies the `discountRate`: 100.0×0.75.