Throughout this section, I've mentioned that a reference variable may only send messages that are available to its type.  In other words imagine that a Person class has a setFirstName method, and an Employee class has a setSalary method:

Person p = new Employee();

// Legal since setFirstName is available to Person

p.setFirstName("Jane");

// Illegal since salary is available

// to an Employee, not a Person.

p.setSalary(80\_000);

I want to be clear, however, that "being available to an object" is not the same as "being declared inside an object."  Let's say we have a subtype of Employee called Instructor.  If we create an Employee reference variable, we can call any member available to Employee - including those members it inherits.  For example:

Employee e = new Instructor();

e.setFirstName("Jane");

... is legal because Employee inherits the setFirstName method from Person.  It wasn't declared inside Employee, but it was available to Employee.  So that's why I specifically say that a reference variable is limited to the members that are available to, rather than defined inside, the class itself.