

For three objects...:

**Student**

**Course**

**Professor**

Choose the attributes that will be **encapsulated** in each of the above objects:

a. List of students taking the course.

b. List of other courses offered by the university.

c. Degree program of the student

d. List of courses the professor teaches.

Correct

You should have identified one attribute for each object:

Student: c. It makes sense for the Student object to keep track of the student's degree program!

Course: a. The course is the most logical place to keep track of all the students taking it!

Professor: d. The Professor object is a great place to keep track of which classes the professor teaches.

Option b does not fit with any of the objects. You might have a full list of courses in a CourseCatalogue object, or something like that!

Methods

**Exposing** methods to be used with other classes thus providing an INTERFACE

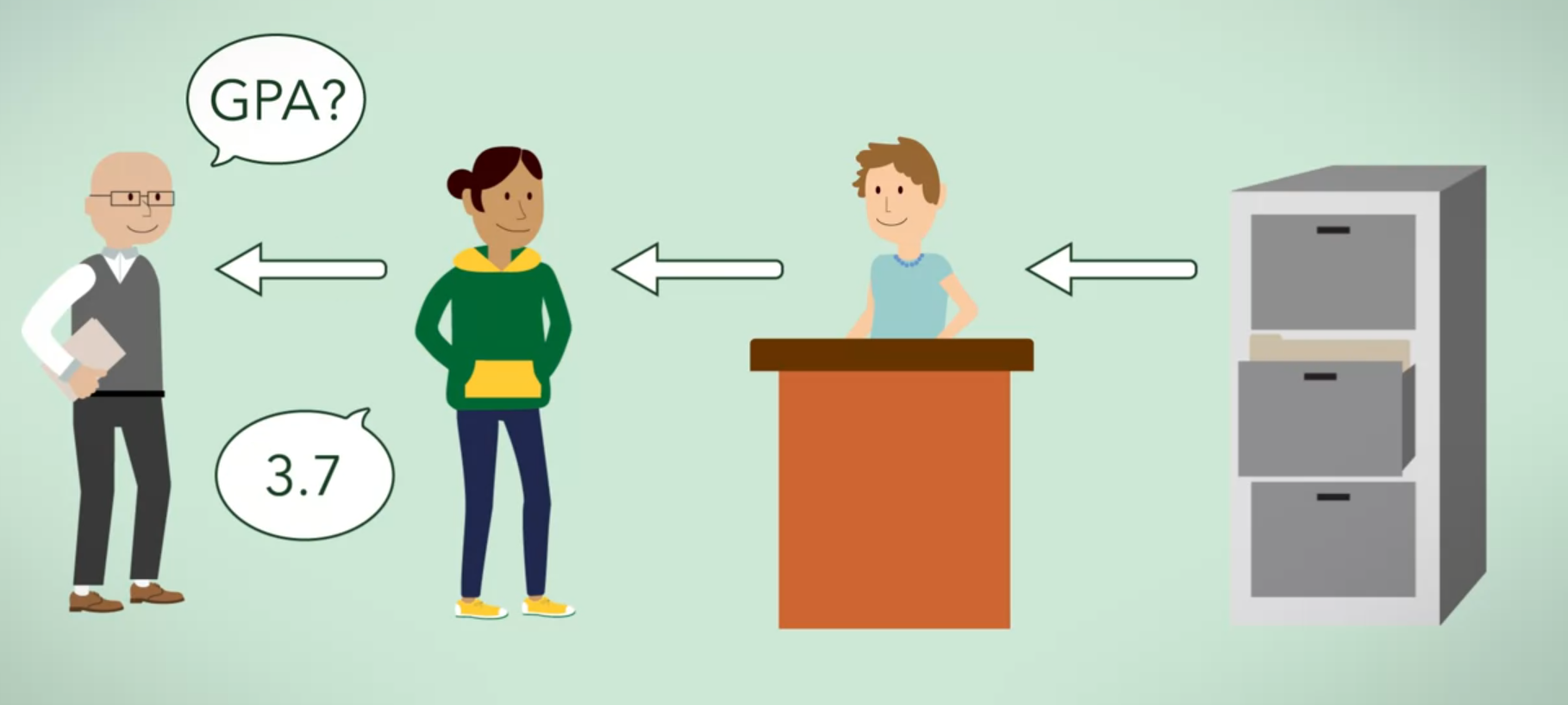




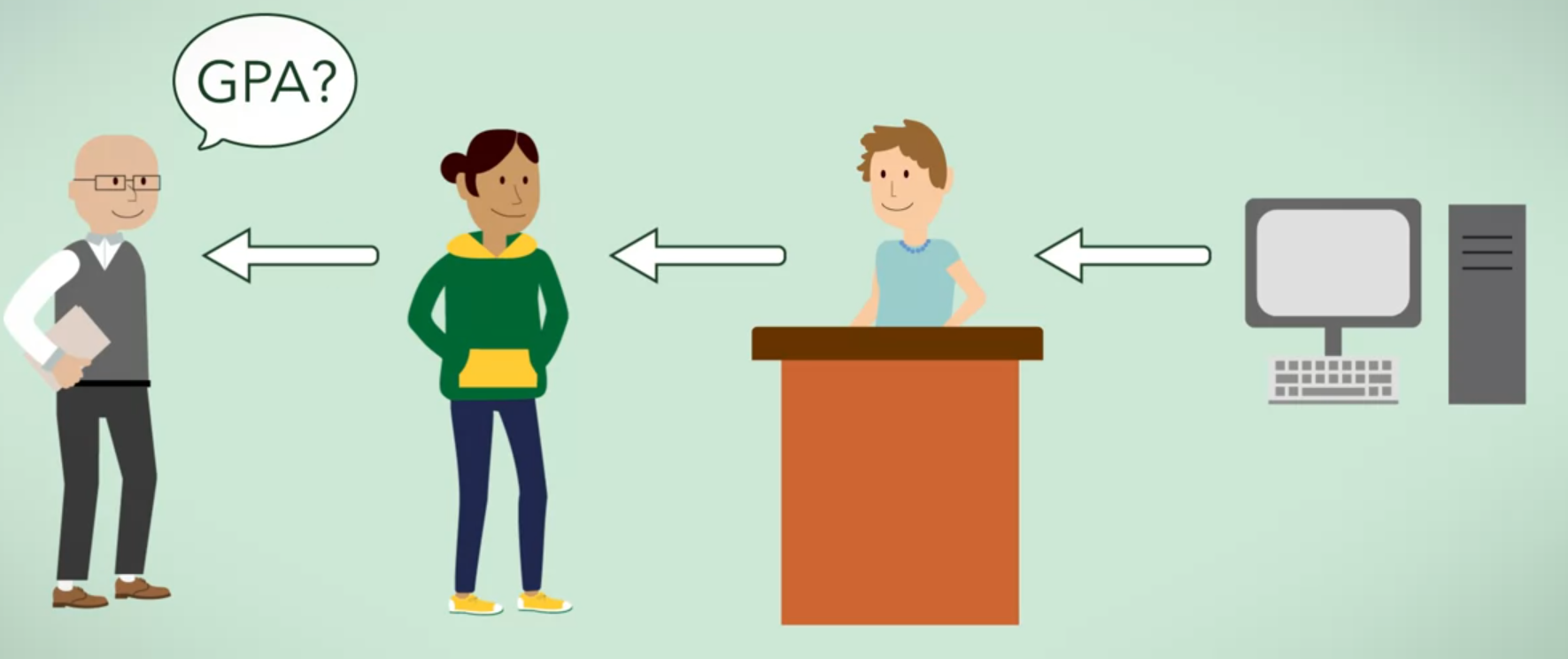


Example: Determining student GPA

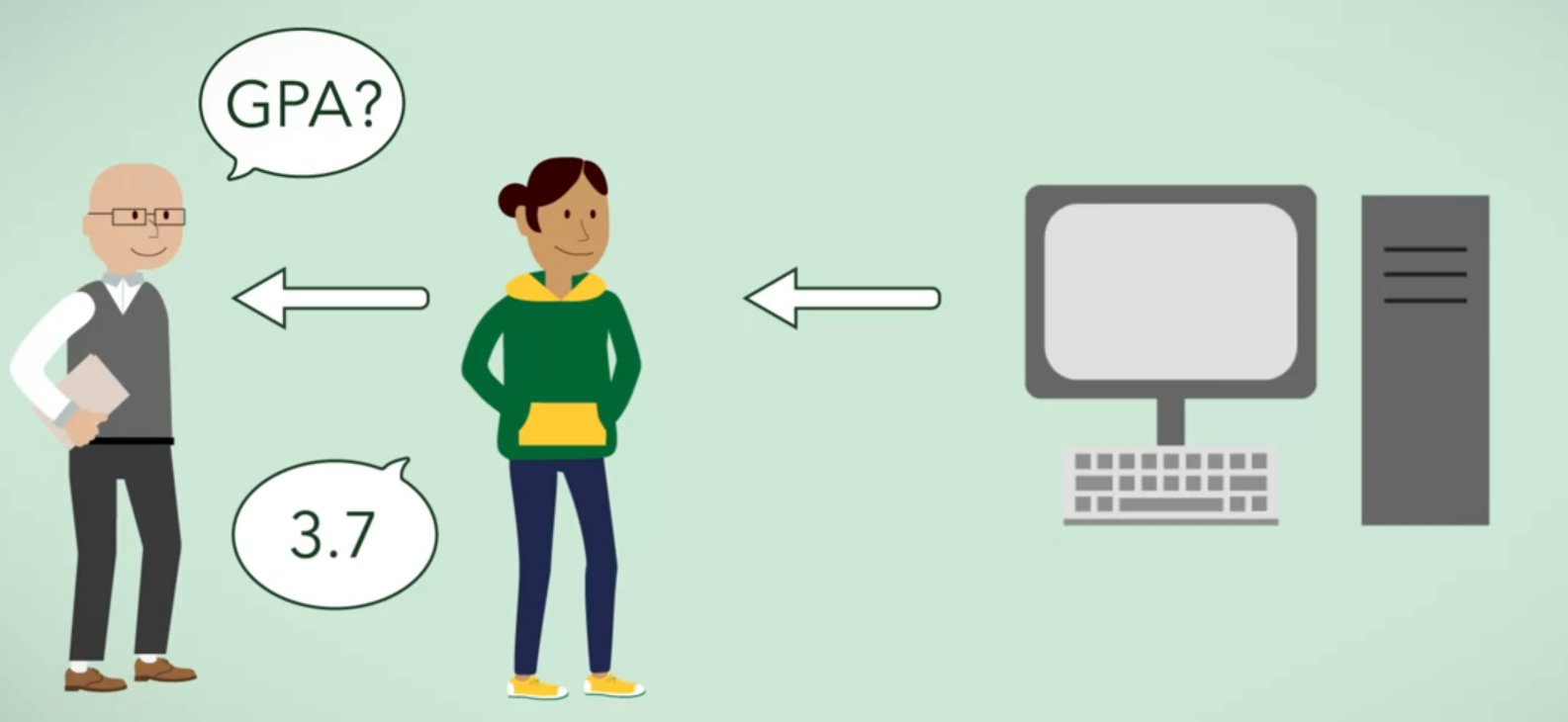
All of these still satisfies the end goal, no matter what the method used. The professor does not care how the student got the GPA!



* Admin getting records from a drawer



* Admin getting records from a computer



* Student directly getting records from student portal





Black Box Thinking

* Doesn’t matter what is going on inside the black box as long as we get the output for our input



* Reduces complexity
* Increases reusability

**What advantages come from encapsulation?**



Attributes of a class are changeable by any other class, so it is easy to change information from anywhere.



Security is increased due to restricted and controlled access to attributes and methods.

Correct

Correct! Since each class is responsible for its own data, it can also control who can change that data.



Reusability is increased because the interface of a class stays the same even if its internal implementation changes.

Correct

That's right! This is called **refactoring** and it will be easier to do with proper encapsulation.



Changing the software is easier because related data and code are located in the same place.

Correct

Great job! It's logical to group data and the methods for manipulating that data in the same place, and classes are a great logical unit for doing that.