## Programmer

Create a class called Programmer. Upon initialization it should receive name (string), language (string), skills (integer). The class should have **two methods**:

* **watch\_course(course\_name, language, skills\_earned)**
  + If the programmer's **language** is **the same as** the **one on the course, increase his skills** with the given amount and return a message **"**{name} watched {course\_name}**"**.
  + Otherwise return **"**{name} does not know {language}**"**.
* **change\_language(new\_language, skills\_needed)** 
  + If the programmer **has the skills** and the **new** **language is not the same as his**, **change** his language to the new one and return **"**{name} switched from {previous\_language} to {new\_language}**"**.
  + If the programmer **has the skills**, but the given **language is equal** to his return **"**{name} already knows {language}**"**.
  + In the last case the programmer does **not have the skills**, so return **"**{name} needs {needed\_skills} more skills**"** and **do not change his language**

Submit only the class in the judge system.

### Examples

|  |  |
| --- | --- |
| **Test Code** | **Output** |
| programmer = Programmer("John", "Java", 50)  print(programmer.watch\_course("Python Masterclass", "Python", 84))  print(programmer.change\_language("Java", 30))  print(programmer.change\_language("Python", 100))  print(programmer.watch\_course("Java: zero to hero", "Java", 50))  print(programmer.change\_language("Python", 100))  print(programmer.watch\_course("Python Masterclass", "Python", 84)) | John does not know Python  John already knows Java  John needs 50 more skills  John watched Java: zero to hero  John switched from Java to Python  John watched Python Masterclass |