

INTERMEDIATE PROGRAMMING ASSIGNMENT

Assignment 1 – Part 1

Background

Imagine that you are working as a software developer in a very prestigious video games company. You have been assigned to a project for a client that wants a video game of Lord of the Rings. Your first task (but not the last, since there will be a second part of the assignment) is to create *orcs* characters.

Exercise

Code the class *Orc*, which represents the orcs of the video game. This class is defined by the following attributes:

1. The orc's name: *name* of type String (for example, "*Ogrorg*")
2. The orc's strength score in the domain [0-5]: *strength* of type Float (for example, 4.3)
3. Does the orc have a weapon? (some orcs own a weapon and others do not own it): *weapon* of type Boolean, which is within the domain [True, False]: (for example, *True*)

Implement the class using the principles of Object-Oriented Programming (marks depend on the right use of these principles). The class must have the following functionalities:

- A constructor for initialising instances
- Properties for accessing and modifying the values of all the attributes.
- Type and value error checking for the attributes of the class. You must check that the values' setting by external users of your module is correct and within the domain of the attributes. For numeric values, if the user introduces a greater value, it will be truncated to the max. value in the domain (for example, the max. *strength* value is 5). If the value is lower than the minimum value, then it will be truncated to the minimum value (for example, the min. *strength* value is 0). For type errors (for example, trying to fix *strength* to "*Ogrorg*"), the assignment will not be completed and the error message "type ERROR" will be printed on the screen.
- The `__str__` method that returns the string representation of the instances in this format: *name strength weapon* (for example, *Ogrorg 4.3 True*). Don't forget the spaces between attributes!
- The overload of the ">" to determine if an *orc* is *greater* than another in terms of fighting. If an orc owns a weapon and the other does not own it, then the orc with a weapon is *greater* than the other one. If both orcs are in the same situation of ownership of weapons, then an orc is *greater* than another only if its strength is greater.
- The orc has a functionality, which is *fight with* another orc. If an orc is ">" than another orc (you must use the overloaded operator for checking such condition), then it wins and its *strength* increases in 1 point. Besides, the instance of the winner orc is printed on the screen (you must use the `__str__` method). Otherwise, that is to say, if there does not exist a ">" orc, they both lose 0.5 points.

Furthermore, you have to create a test script for checking all the functionalities of the orc class. You have to import the module *orc* and test each of the above-described functionalities (marks depend on the right functionality of your class).

Assignment 1 – Part 2.

To be Continued....