CS Homework #18
OOP: Classes, part II

Deadline: 2/29/2020, 9:00 pm. Save your code as lastname_homework18.py and submit on Edmodo. Please, run your code before submitting. If you get an error, try to fix it before submitting your homework. If you get help from anyone, please, make sure that you actually understand the solution.

Task 1 (creating a class, instance initialization __init__, instance attributes)

We continue developing an RPG game prototype.
Create a class Monster with four attributes: unique ID, health points (HP), power, and level. The ID, HP, and power should be an argument passed by the user. The initial level is always 1.

Task 2 (instance method printing information about the instance attributes)

Add a basic introduction for the monster, which should look something like this: “Monster 101, 6 hp, 4 power, and level 1.”

Task 3 (class attributes; using class attributes to update instance attributes)

Add two *class* attributes: (1) time, which is a string that can be either “day” or “night”, and (2) bonus, which is an integer that can be either -1 or 2. This is a bonus to the monster power: -1 during the day and +2 during night. Start with initial values: time = “day” and bonus = -1.

Add a simple *instance* method which adds the value of the bonus to the power of the monster.

Task 4 (class method updating class attributes)

Implement a *class* method (not to be confused with an instance method) to change the time to either “day” or “night”. The method should accept a string argument from the user, “day” or “night” AND update both class attributes from task 3, *time* and *bonus* (the value depends on whether it is day or night as described above).

Task 5 (instance method updating instance attributes)

Implement an *instance* method that adjusts the level of the monster on the basis of the level of the hero and difficulty level. Therefore, there are two arguments: an integer value which is the level of the hero and a string that can be either “easy”, “normal”, or “hard”. The level of the monster is equal to the level of the hero. It is further adjusted given the difficulty. For “easy” it decreases by 2, for “hard” it increases by 2 (and no change for “normal” difficulty”).

Task 6 (generating multiple instances; using random number generator)

Generate 10 monsters using a for loop and random values for HP (between 1 and 9) and power (between 3 and 6). Notice that the unique ID values should start at 101 and then increase for each new monster (102, 103, and so on; hint add 101 to i from the “for i in range” loop). The instances should be saved as elements of a list.

Display information for the *first* monster in the list using the introduction method from Task 2. Now, choose a *random* monster and display its information.

Task 7 (iterating over the list of instances; applying class and instance methods)

First, use for loop together with the introduction (from Task 2) to display information about each monster. Second, use the class method to change the time to “night” (the class method from Task 4, which should also update the bonus value). Third, use for loop to update the power of each monster given the new power bonus (the instance method from Task 3). Finally, display introduction for each monster again to verify that the power increased for everyone.

Task 8 (iterating over the list of instances and applying an instance method)

Now assume that the monster is meeting a hero who is level 50 playing on “hard” difficulty. Use for loop and the instance method from Task 5 to adjust the level of each monster*.

Verify that each monster in the game is now level 52.

* Note: In principle, this particular task can be solved by modeling difficulty as a class attribute. However, I wanted you to practice applying instance methods while iterating over the list of instances. Moreover, one can imagine that if multiple heroes are playing the game then different monsters will have to be scaled differently depending on the level of the heroes whom they meet.