CS Homework #20
OOP (inheritance and polymorphism)

Deadline: 3/21/2020, 9:00 pm.
*Save your code as lastname_homework20.py and submit on **Edmodo**.*

Task 1
We continue developing parts of an RPG game. Create a class describing an 'inventory item'. Each inventory item must have the following attributes specified by the user:

1) Condition, which is a number between 0 (completely broken) and 100 (brand new),
2) Rarity, which is a string: "common", "rare", "epic".

In addition, each inventory item has

3) monetary value in coins. Initially, you can set it equal to zero.

Task 2
Implement an instance method which updates the monetary value of the item, given the following formula: Coins value = condition × "rarity coefficient", where the rarity coefficient is equal to 1 for common items, 2 for rare, and 3 for epic.

Update your def __init__ method with a call to this method to calculate the coin value of the item.

Task 3
Implement an instance method which upgrades rarity of the item ("common" becomes "rare"; "rare" becomes "epic") and then calls the "update value" method (from Task 2) to calculate the new value of the item in coins.

Task 4
Create an instance method that displays the rarity and condition of the item. Optional: try to use self.__class__.__name__ to display the type of the item (see class notes for an example). Create an inventory item object and verify that displayed information is correct.

Task 5
Create two inventory child classes (or subclasses): "sword" and "shield", using "inventory item" class above as the parent class, or superclass.

Each subclass has a unique *class attribute*:

- base_attack = 10 for swords,
- base_defense = 8 for shields.

Task 6
Implement *instance methods* for “sword” and “shield” classes, which calculate the *actual* attack value (for swords) and *actual* defense value (for shields), using the formulas:

For swords:

Actual attack = base attack * rarity coefficient * (Condition/100).

For shields:

Actual defense = base defense * (rarity coefficient + 1) * (Condition/50).

Task 7
For both subclasses, create instance methods that displays the rarity, condition, and actual attack/defense values of the items. Use the same name for you instance as in Task 4 (yes, you are overriding the method!)

Task 8
Create a hero inventory, consisting of two swords and two shields. Use random number generator to generate the initial conditions and rarity. Display information about each item in the inventory using the methods from Task 7 (polymorphism).

Task 9
Assume that there was long battle and the conditions of one of the swords and one of the shields decreased by 50%. Use the previously created methods to re-calculate: (a) the monetary value of the items, and (b) actual attack/defense values for the items. Again, display all information about each item in the inventory.