Object Oriented Programming, Exercise 2 (1 point from each task)

Topics: Increase coding routine in Python, using Git, pseudocode, a bit of classes and objects

Make a Git commit at least after every coding task.

Code in Python3 and follow the style guide.

1. Explain the following terms:
    a. Pseudocode
    b. Algorithm
    c. Data attribute
    d. Method
2. Take a look at the course's assessment (points from exercises meaning certain grade). Write **pseudocode** for a program where user inputs the exercise points and program prints out the grade.
3. After writing the pseudocode, code task 2. Simple code is enough, no objects needed. Use informative and readable output prints.
4. Write **pseudocode** for a program that accepts student's name and grade as input and counts the average of grades of all students. If you have difficulties, you can fix the number of students to e.g. 5. Print out the average.
5. After writing the pseudocode, code task 4. Simple code is enough, no objects needed. Use informative and readable output prints.
6. Take a look at the coin.py, write it down in your IDE and run it. See that coin gets tossed.
7. Modify the toss_the_coin() function so that there are 3 more options: Coin lands on the table upright (and not flat showing heads or tails), coin drops on the ground and disappears on a rabbit hole or coin defies gravity and gets lost on a wormhole in space. Name the options properly and give informative and readable output of the status.
8. Imagine you would have to code a **simple** alarm clock (shows time and alarms you at certain time you can set). Which data attributes will you have? Do the attributes have some value restrictions? You should find at least 5 data attributes. Which methods would you need? Which methods should be public and which ones should be private?
9. Write pseudocode for the alarm clock (see task 6). Notice, this is a *simple* alarm clock. This is a simple task, do not make it unnecessarily complicated! For example, you can start with a simple clock without the alarm. If this gets too complicated, write pseudocode for a timer instead of an alarm clock (each round in a loop increases the seconds with 1 etc.).
10. Code the alarm clock, use objects. This is also supposed to be a simple task, so do not make it unnecessarily complicated. Alarm can simply be a text output on screen. No sounds are needed (can be added if you wish). If this is too difficult, code a timer instead.

Return and demo

1. Return your code, pseudo code and test report to Itslearning by deadline.

2. Demonstrate your code and pseudocode to teacher on-site (before deadline or within two weeks after it).

Pseudocode:

https://www.geeksforgeeks.org/how-to-write-a-pseudo-code/