

Object Oriented Programming, Exercise 5

Topics: Passing objects as arguments, storing objects in a list, storing objects in a dictionary. You get 1 point from each task unless otherwise specified.

Make a Git commit at least after every coding task.

Code in Python3 and follow the style guide.

1. Multiple choice:

- a. The _____ method is automatically called when an object is created.
 - i. `__init__`
 - ii. `init`
 - iii. `__str__`
 - iv. `__object__`
- b. The _____ programming practice is centered on creating functions that are separated from the data that they work on.
 - i. modular
 - ii. procedural
 - iii. functional
 - iv. object-oriented
- c. The _____ programming practice is centered on creating objects.
 - i. object-centric
 - ii. objective
 - iii. procedural
 - iv. object-oriented
- d. A(n) _____ is a component of a class that references data
 - i. method
 - ii. instance
 - iii. data attribute
 - iv. module
- e. By doing this, you can hide a class's attribute from code outside the class.
 - i. avoiding using the self-parameter to create the attribute
 - ii. begin the attribute's name with `private__`
 - iii. begin the name of the attribute with two underscores
 - iv. begin the name of the attribute with the symbol `#`
- f. A(n) _____ method stores a value in the data attribute or changes its value in some other way.
 - i. modifier
 - ii. constructor
 - iii. mutator
 - iv. Accessor

2. Explain the following terms:

- a. Super class

- b. Sub class
 - c. Base class
 - d. Derived class
 - e. "Is a" relationship
- 3. Create multiple dices (at least three) and put them in a list. See that your dice can be rolled and the side can be shown. Create a small game where the best sum of three rolls wins. If the sum is a tie, tied dices are rolled as long as a winner is found (best side wins). Use functions and pass objects (or list of objects) as arguments. Use informative and clear output prints. (1 point)
 - a. Make your solution to be usable for any number of dices, e.g. number of dices is asked from the user. (1 point)
- 4. Create a class called Player. Player has at least the following data attributes: first name, last name and a player id. Remember to code accessor and mutator methods and str-method. Create a dictionary so that the player id is a key and each player has one dice. Roll each player's dice and print out each player's dice's side. Use informative and clear output print.
- 5. Create a class called Student and use the following data attributes: first name, last name and student id. Remember to code accessor and mutator methods and str-method. Store students and their pet mammal to dictionary (use the mammals from Exercise 4). Think, what should be used as the dictionary key. Code a function that prints out each student and their mammal's information. Use informative and clear output print.
- 6. Let the student select his/her pet mammal by rolling 2 dices and using their sum to indicate a correct mammal. The higher the number, the heavier mammal you get as your pet. (2 points)
- 7. Take the code main.py from Itslearning, **do not change it** (except for the Author and code after line 42; obviously for testing purposes you can e.g. comment the methods out that you have not yet implemented, but in your final return the main.py shall be like given). Implement classes **Card** and **Deck** (in their own modules) so that the main.py can be run and the output is exactly the same than in file Ex5_task7_output.txt (of course, shuffled deck and drawn cards can be different). (You can look for help here: <https://medium.com/@anthonytapias/build-a-deck-of-cards-with-oo-python-c41913a744d3>). Then implement one of the games described below (or even all of them if you like). (2 points)
 - a. Draw 3 cards, highest value wins. Announce results (have clear output print). Have a re-draw if there are ties.
 - b. Implement card game Twenty-one (= Ventti in Finnish) or Blackjack for as many players as you like. Announce results clearly. Notice, you do not necessarily need a Player class in this game (but you are allowed to have it).
 - c. Create a Player class as well. Each player shall have a name and hand of cards. Then implement any card game you like (an existing one or create your own). Comment the code clearly and explain the rules as well. Have clear output prints.