

## Object Oriented Programming, Exercise 4

**Topics: Modules and modularity, accessor and mutator methods, state of the object**

**Make a Git commit at least after every coding task.**

**Code in Python3 and follow the style guide.**

1. Explain the following terms and what they are used for:
  - a. Inheritance (in object-oriented programming)
  - b. Multiple inheritance
  - c. UML
  - d. UML class diagram
2. True or false?
  - a. The practice of procedural programming is centered on the creation of objects.
  - b. Object reusability has been a factor in the increased use of object-oriented programming.
  - c. It is a common practice in object-oriented programming to make all of a class's data attributes accessible to statements outside the class.
  - d. Class methods do not have to have a self-parameter.
  - e. Starting an attribute name with two underscores will hide the attribute from code outside the class.
  - f. You cannot directly call the `__str__` method.
3. Answer the following question: When you model using UML diagrams, why is it important to follow the UML syntax strictly?
4. Take the cell phone class of last week and divide the main function into another file (name the file clearly, e.g. `main.py`). Leave the cell phone class in the original file. Test, that your code still works.
5. Add an ID data attribute (integer between 1-6) to the cell phone. Cell phone class shall have accessor and mutator methods for all data attributes. Also check the `__str__` method is up to date.
6. Create different cell phone objects (which have different data attribute values, use mutator methods to change the data attribute values). Print out each object's state (use the `__str__` method in the cell phone class). Use a loop with consequent IDs. Remember to add phones to a list. You'll need the list in task 7.
7. Take the Dice class from your earlier exercises and place that to its own file. Then in main function roll a dice and based on the result choose the correct cell phone based on the ID. Print out the chosen cell phone object's state.
8. Create a car object. It has the following data attributes: make, model, mileage, price, color, maximum load limit, size of trunk. Make them private. Write accessor and mutator methods to change them. Add `__str__` method to print the state of the car.
9. Create a mammal object. It has the following data attributes: ID, species, name, size and weight. Roll the dice, based on the result check if the correct mammal (based on ID) fits into your car's trunk (that you created in previous task). Also check that your mammal

does not exceed the car's load limit. Give informative output prints and error messages.(2 points)