MyRepos\KI-Kurs-Mystro\Exercise_26062024.py

```

"""
# Exercise at 26.06.2024 from Eng. Ola
- Class and 'Objects
- constactor in class:    def__init__(self):    # es gibt auch destructor
- Define function in class:    def mysum(self, x, y):
"""

"""
# Task 01
1. Define the Class:
○ Create a class named Car.
○ The class should have the following attributes: make, model, year, and
odometer_reading.
2. Constructor Method:
○ Define an __init__ method to initialize these attributes. The
odometer_reading should be initialized to 0.
3. Methods:
○ Define a method named get_description that returns a neatly formatted
descriptive name for the car.
○ Define a method named read_odometer that prints a statement showing the
car's mileage.
○ Define a method named update_odometer that sets the odometer reading to a
given value. This method should reject any attempt to roll back the odometer.
○ Define a method named increment_odometer that increments the odometer
reading by a given amount.
"""
class Car:                                  # Create a class Car
  def __init__(self, make, model, year):    # define the function __init__
    self.make = make
    self.model = model
    self.year = year
    self.odometer_reading = 0

  def get_description (self):                  # function get_description
    #print(f" {self.make}  {self.model}  {self.year}")
    return  (f"My car is  {self.make}  {self.model}  {self.year}")

  def read_odometer(self):                    # function read_odometer
    #print(f"my car has {self.odometer_reading} km.")
    return (f"My car has {self.odometer_reading} km.")

  def update_odometer(self,km):               # function update_odometer
    if km >= self.odometer_reading:
      self.odometer_reading = km
    else:
      print("km less than the old km")

  def increment_odometer(self,km):            # function increment_odometer
    self.odometer_reading += km

def main():

    #  make Object from Car:
    mycar = Car("Nissan", "Qashqai", 2018)

    print(mycar.get_description())
```

```
    print(mycar.read_odometer())
    mycar.update_odometer(55000)
    print(mycar.read_odometer())
    mycar.increment_odometer(1000)
    print(mycar.read_odometer())

if __name__ == "__main__":
    main()


"""
# Task  02
1. Define the Class:
o Create a class named Dog.
o The class should have the following attributes: name and age.
2. Constructor Method:
o Define an __init__ method to initialize these attributes.
3. Methods:
o Define a method named sit that prints a message indicating the dog is sitting.
o Define a method named roll_over that prints a message indicating the dog is
rolling over.
"""
class Dog:                      #Create a class Dog
  def __init__(self,name,age):  # class with name and age
    self.name = name
    self.age = age

  def sit(self):          # function sit
    print(f"{self.name} is sitting")

  def roll_over(self):    # function roll over
    print(f"{self.name} is rolling over ")

def main():

    #  make Object from Class Dog:
    mydog = Dog("Rambo", 10)    # call class Dog in omject mydog

    mydog.sit()               # call the sit function
    mydog.roll_over()         # call the roll over function

if __name__ == "__main__":
    main()


"""
#  Task 03
Task3: Create a Student Class
1. Define the Class:
o Create a class named Student.
o The class should have the following attributes: name and courses.
2. Constructor Method:
o Define an __init__ method to initialize the name attribute and initialize
courses as an empty list.
3. Methods:
o Define a method named enroll that takes a course name as a parameter and
appends it to the courses list.
o Define a method named get_courses that prints the list of courses the student
is enrolled in.
"""
```

```
118
119 class Student():
120   def __init__(self,name):
121     self.name = name
122     self.courses = []
123     print(f"Student  {self.name}  is created")
124
125   def enroll(self,course_name):
126     self.courses.append(course_name)
127
128   def get_courses(self):
129     print(f"{self.name} has this courses  {self.courses}")
130
131 def main():
132
133     #  make Object from Clss Student:
134     st1 = Student("Attia")
135     st1.enroll("Matha")
136     st1.enroll("English")
137     st1.enroll("KI-Kurs")
138
139     st1.get_courses()
140
141 if __name__ == "__main__":
142     main()
143
144
145
```