Classes and Objects

Week 3

Classes

- Python is an Object Oriented Language which means that the code can be divided into individual code, namely objects.
- Each of these objects is an instance of a so-called class.
- Everything that is indented after the colon belongs to the class.
- Class Car:
    - def __init__(self,manufacturer,model,hp):
        - Self.model = model
        - Self.hp = hp
        - Self.manufacturer = manufacturer

Constructor

- Is a special function __init__ called as constructor
- Every time we create an instance or an object of our class ,we use this constructor.
- It accepts parameter
- First one is self which is mandatory.

Adding Functions and Variables to the class

- Functions
  - Simply add functions to our class that perform certain actions.
  - These actions can also access the attributes of the class.
- Variables
  - Class Car:
    - amount_cars = 0

```
Def __init (self ):
    amount_cars +=1
```

Every time a new class of car is created then the amount_car is increased by 1.

Destructors

- In Python, we can also specify a method that gets called when our object gets destroyed or deleted and is no longer needed. This is called as destructor and it is opposite of the constructor.
- def__del__(self):

Creating Objects

- Now we have implemented the class, we can start to create some object of it.
- Mycar1 = car(“Tesla”,”Model X” ,525)
- MyCar2 = car(“BMW”,”X4”,355)

INHERITANCE

- Use existing classes and to extend them with new attributes and functions.
- Same function in different classes with same parameters.

Operator Overloading

- Operator overloading in Python allows you to define how operators like +, -, *, /, etc., behave with objects of your custom classes. This gives you the ability to customize the behavior of these operators based on the semantics of your class