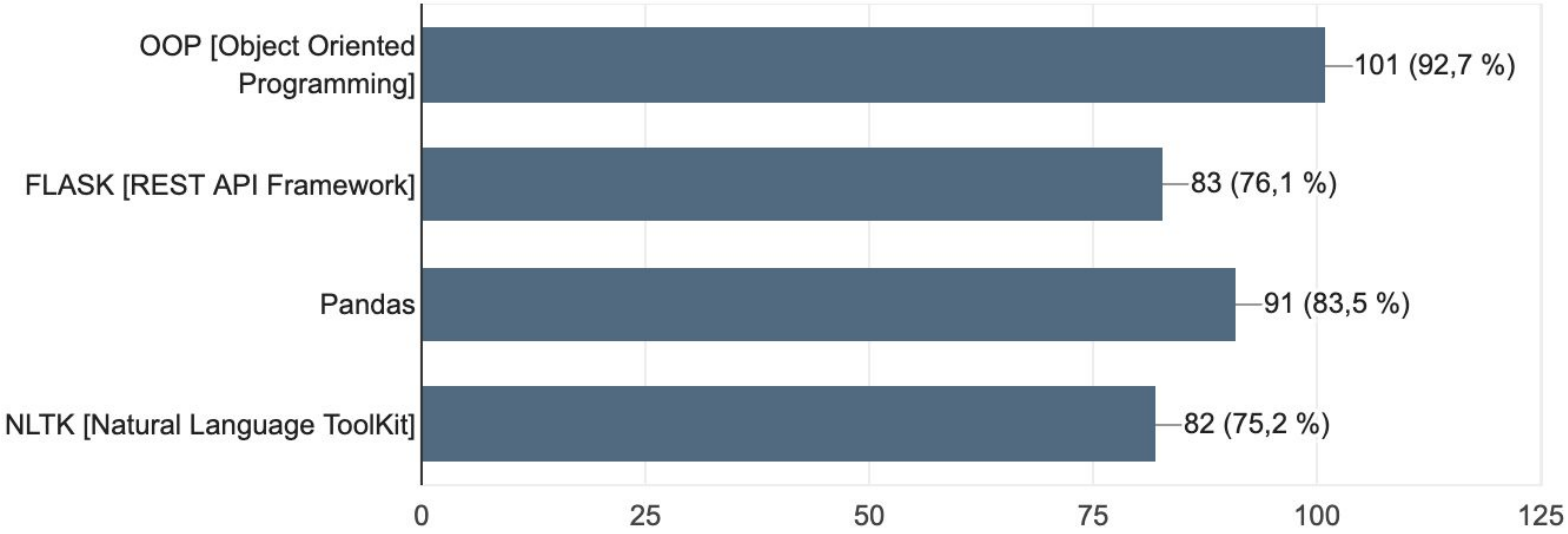
IEEE OOP with Python

16 June 2020 (Tuesday)

109 Antworten



Conclusion from survey/ future courses

16 June 2020 (Today): OOP

23 June 2020*: Pandas

30 June 2020*: FLASK

**** Dates are tentative***

Motivation

- If you write a ***big program*** (code), things will start looking messed up.

- ***Unstructured*** & ***lengthy*** code can slow down the development process.

- Reduced ***flexibility*** for changes.

To avoid above one should use object oriented programing.

Definition of OOP

Not lo

Article Talk Read Edit View history

Wiki Loves Earth 2020 photo competition: take photos in nature and support Wikipedia.

Object-oriented programming

From Wikipedia, the free encyclopedia

"Object-oriented" redirects here. For other meanings of object-oriented, see Object-orientation.
"Object-oriented programming language" redirects here. For a list of object-oriented programming languages, see List of object-oriented programmin

Object-oriented programming (**OOP**) is a programming paradigm based on the concept of "objects", which can contain data, in the form of fields (often known as *attributes* or *properties*), and code, in the form of procedures (often known as *methods*). A feature of objects is an object's procedures that can access and often modify the data fields of the object with which they are associated (objects have a notion of "this" or "self"). In OOP, computer programs are designed by making them out of objects that interact with one another.[1][2] OOP languages are diverse, but the most popular ones are class-based, meaning that objects are instances of classes, which also determine their types.

Many of the most widely used programming languages (such as C++, Java, Python, etc.) are multi-paradigm and they support object-oriented programming to a greater or lesser degree, typically in combination with imperative, procedural programming. Significant object-oriented languages include Java, C++, C#, Python, R, PHP, JavaScript, Ruby, Perl, Object Pascal, Objective-C, Dart, Swift, Scala, Kotlin, Common Lisp, MATLAB, and Smalltalk.

Object Oriented Programing

When you program things but with reference to a common entity.

That entity is object, which is derived from a class.

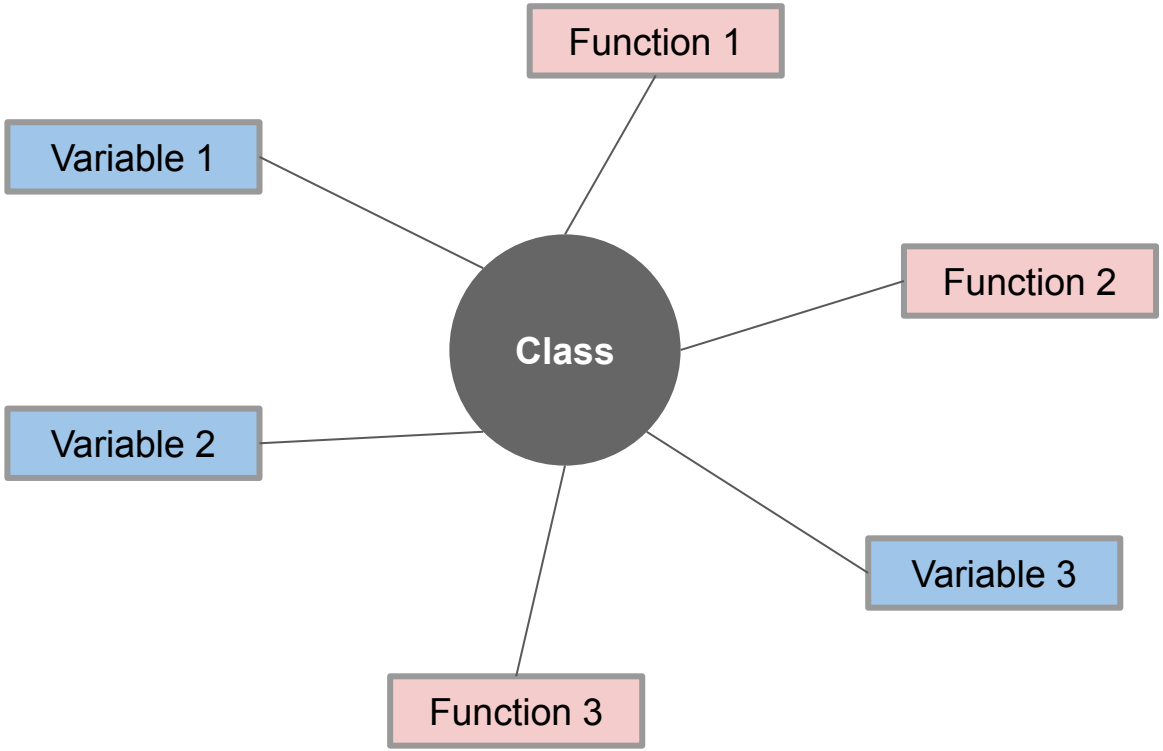
What is class?

Class is a bunch of specifications.

These specifications consist of variables, functions (with different rights), constructor, destructor and many ***other routines***.

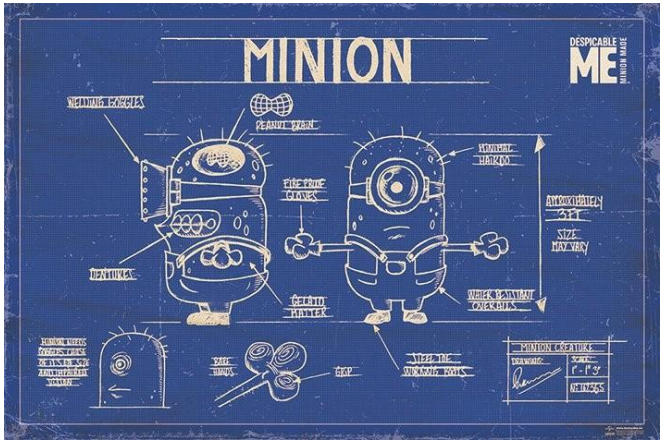
Other routines are default functions which your can overwrite.
Example: Constructor

Visualization



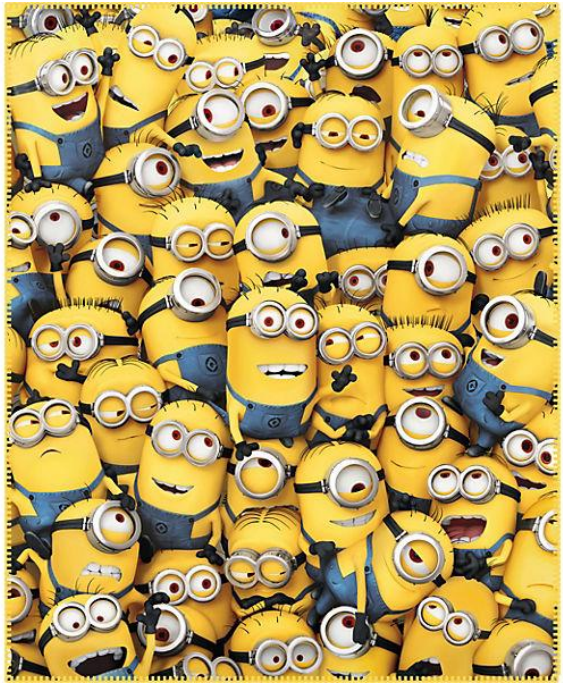
What is object ?

Class (Blueprint)



1:N

Objects



The most simple class

```
Class <CLASS_NAME>:

    pass
```

Object creation is like creating a copy of class with its specifications.

Example:

```
Class Human:

    Name= None,

    Age= None
```

```
object = Human()

object.Name= “Bob”,

object.Age= “25”
```

Object is a variable.

```
class Human:
    Name: None,
    Height: None,
    Weight: None,
    Age: None
    def sleep( ):
        print("Human is sleeping")
    def eat( ):
        print("Human is eating")
```

Variables

Functions

Default functions

Constructor: __init__(self): *returns nothing*

Destructor: __del__(self): *returns nothing*

Printing string: __str__(self): *returns string*

Echo string: __repr__(self): *returns string*

Iterable: __iter__(self): *returns iterable object*

EAT

SLEEP

CODE

REPEAT