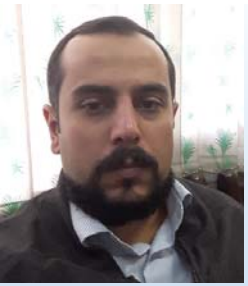

Advanced Course of Python Programming Language:
"Object Oriented Programming"

Lecturer: Eng. Abolfazl Mohammadijoo

www.abolfazlm.com

Object Oriented Programming

Note:

This lecture includes about 100 slides and I just provide first 10 pages here. I added some exercises, homework and quizzes to furnish course better to understand. Also, there are some other sources which I used in this lecture, as below:

Ebook: Learning Python: Powerful Object-Oriented Programming, 5th Edition, O'Reilly, by "Mark Lutz"

Object Oriented Programming

Purposes and Outcomes of this Lesson

1- Design reusable Object-Oriented Python Classes
2- Apply powerful OOP concepts to handle complexity
✓ Classes, instances
✓ Encapsulation, inheritance, polymorphism
3- Handle errors (exceptions)
4- Serialize (store) objects for later use
5- Debug, test and benchmark your code

Object Oriented Programming

6- Learn OOP concepts used across many language:

Java, C++, JavaScripts, et al.

7- Be able to contribute python code on a professional level

8- Understand OOP terminology when discussed online or in interviews

9- Take an essential next step in your python education

10- if already familiar with OOP, see how it is applied in "Pythonic" way

Object Oriented Programming

OBJECT-ORIENTED PROGRAMMING (OOP)

- Developed in the 1960's (Simula67, Smalltalk)
- A paradigm for code organization and design
- The OOP Paradigm:

✓ Organizes data into objects and functionality into method

✓ Defines object specifications (data and methods) in classes

- Promote Collaboration, code extension and maintenance
- The primary paradigm for software design worldwide

Object Oriented Programming

Procedural vs. Object Paradigm

Procedural Paradigm

```
this = 0
this = increment(this)
this = increment(this)
print(this)          #2
```

Object Paradigm

```
this = MyCustomInt()
this.increment()
this.increment()
print(this)          #2
```

(**this** is **MyCustomInt** object)

One definition of *object*: a unit of *data* that has associated *functionality*

Object Oriented Programming

Procedural vs. Object: Library Code

(this is for reference – we will discuss about details later)

Procedural Paradigm

```
def  increment(arg):
    arg = arg + 1
    return arg
```

Object Paradigm

```
class MyCustomInt(object):
        def  __init__(self):
            self.val = 0
        def  increment(self):
            self.val = self.val + 1
        def  __repr__(self):
            return  str(self.val)
```

Object Oriented Programming

OOP: WHY?

- **OOP organize the code so it is:**
  - Easier to use
  - Easier to understand
  - Easier to maintain and extend
  - Easier to collaborate
- **Complexity must always be managed**
- **OOP is universal paradigm (many languages)**
- **Learning OOP is a necessary next step into the larger world of software engineering**

Object Oriented Programming

OOP: three Pillars

- **Encapsulation**
- **Inheritance**
- **Polymorphism**

Object Oriented Programming

Object-Oriented Python

- Everything is an object, even number
- Other Languages employ primitives (non-object data)

What is an Object

- An object is a unit of data (having one or more attributes), of a particular class or type, with associated functionality (methods)

WWW.ABOLFAZLM.COM

THANK YOU FOR YOUR ATTENTION!

You can keep in touch with me for any other possible helps or workshops, via:

Emails: a.mohamadijoo@gmail.com & info@abolfazlm.com

Mobile No: 09124908372 & 09365388409