



# Second Semester Class 3

6/08/2023

Standup



Github



# Homework Discussion





# Object Oriented Programming in Python

Object Oriented Programming (OOP) is a way for developers to create objects within their code that have defined methods and attributes. OOP is a great way to improve scalability, organization and effectiveness of your code.

- **Classes**
  - Easy way to bundle data and functionality together.
  - When creating a class, a new type of *object* is created, which in turn can allow new instances to be made.
- **Attributes**
  - Instances that can either be shared across every object of the class, or instances that can be unique to each object of the class.
- **Methods**
  - Functions defined inside the body of a class
  - Easiest way to think of these is to consider them as functions that take the object into account through use of the self argument
  - Think of it as :
    - Class = Object, or set of objects
    - Attribute = Characteristics of Object, or Class
    - Method = Actions the obj

# Hands on with Classes, Attributes and Functions



# Homework

- 1 Hour of Udemy, Complete Python Bootcamp Zero to Hero Section 8
- Complete the following reading assignments:
  - [In Depth w/ Attributes](#)
  - [In Depth with Methods](#)
- [OOP Exercise 1](#)
- Create a Class (ANYTHING you want):
  - Ensure your class has *BOTH*:
    - Class specific Attributes
    - Object specific Attributes
      - For the object specific attributes ensure you ensure that the variable types are specified (ie - I cannot put a string where the answer would be expect to be a integer)
  - Create 2-3 methods for your class
    - Ensure that you use methods that require self as well as a user specified variable.
- Create a Python class called SalesPerson that has the following:
  - Dynamic variables called firstName and lastName that provide the salesperson's first and last name
  - A static variable (starts with 0) called sales, to track their sales over time.
  - A method called makeSale that takes an *integer* and adds that integer to sales.
  - A method called salesReport that prints a string saying "My total sales are {sales}!"
    - Once you have all of this, create a SalesPerson instance with firstName = 'Annie' and lastName = 'Mae' and have them make a sale of 250, and run the salesReport method.
    - Create a second SalesPerson instance with firstName = "Manda" and lastName = "Lynn" and have them make a sale of 150.
    - Create a third SalesPerson instance with firstName = "Dick" and lastName = "Biscuits" and have them make a sale of 6969.
  - BONUS create a method that will print all instances of SalesPerson in order of their sales made greatest to least.
- Upload the last three bullet points to your HW repositories in Github.