**Python\_Lesson3: Object Oriented Python**

Please don't forget to submit your feedback after the class.

**Lesson Overview:**

In this lesson, we will review Object Oriented Python and NumPy package.

Classes are one of the important concepts of Python. Everything in Python is an Object. Classes enable us to encapsulate data, restrict the scope of data members and functions. They help us in reusability by inheritance. We can define the various level of data encapsulation like private, protected and public.

**Use Case Description:**

1. Inheritance (Bank Account)

2. Multiple Inheritance (Clock and Calendar)

3. numpy

4. web scraping

**Programming elements:**

Object Oriented concepts (classes, constructors, inheritance etc.)

**In class programming:**

1. Create a class Employee and then do the following

a. **Create a data member to count the number of Employees**

b. **Create a constructor to initialize name, family, salary, department**

c. Create a function to average salary

d. Create a Fulltime Employee class and it should inherit the properties of Employee class

e. Create the instances of Fulltime Employee class and Employee class and call their member functions.

**2. Web scraping**

Write a simple program that parse a Wiki page

mentioned below and follow the instructions:

https://en.wikipedia.org/wiki/Deep\_learning

a. Print out the title of the page

b. Find all the links in the page (‘a’ tag)

c. Iterate over each tag(above) then return the link using attribute "href" using get

**3. Numpy**

Using NumPy create random vector of size 15 having only Integers in the range 1-20.

Then reshape the array to 3 by 5

Then replace the max in each row by 0

(you can NOT implement it via for loop. You need to use **np.where, reshape**)

**Evaluation Criteria:**

1. Completeness of Features

2. Code Quality (<https://en.wikipedia.org/wiki/Best_coding_practices>)

3. Time

4. Feedback Submission

**Note:** *Cheating, plagiarism, disruptive behavior and other forms of unacceptable conduct are subject to strong sanctions in accordance with university policy. See detailed description of university policy at the following URL:* [*https://catalog.umkc.edu/special-notices/academic-honesty/*](https://catalog.umkc.edu/special-notices/academic-honesty/)

'''Generate A Random Number From The Normal Distribution  
np.random.normal()  
0.5661104974399703'''  
'''Generate Four Random Numbers From The Normal Distribution  
np.random.normal(size=4)  
array([-1.03175853, 1.2867365 , -0.23560103, -1.05225393])'''  
'''Generate Four Random Numbers From The Uniform Distribution  
np.random.uniform(size=4)  
array([ 0.00193123, 0.51932356, 0.87656884, 0.33684494])'''  
'''Generate Four Random Integers Between 1 and 100  
np.random.randint(low=1, high=100, size=4)  
array([96, 25, 94, 77])'''

**numpy.nanmax**

**numpy.nanmax(***a***,***axis=None***,***out=None***,***keepdims=<no value>***)**[**[source]**](https://github.com/numpy/numpy/blob/v1.17.0/numpy/lib/nanfunctions.py#L344-L452)

Return the maximum of an array or maximum along an axis, ignoring any NaNs. When all-NaN slices are encountered a RuntimeWarning is raised and NaN is returned for that slice.

**numpy.amax**

**numpy.amax(***a***,***axis=None***,***out=None***,***keepdims=<no value>***,***initial=<no value>***,***where=<no value>***)**[**[source]**](https://github.com/numpy/numpy/blob/v1.17.0/numpy/core/fromnumeric.py#L2504-L2621)

Return the maximum of an array or maximum along an axis.

|  |  |
| --- | --- |
| **Parameters:** | **a : *array\_like***  Input data.  **axis : *None or int or tuple of ints, optional***  Axis or axes along which to operate. By default, flattened input is used.  *New in version 1.7.0.*  If this is a tuple of ints, the maximum is selected over multiple axes, instead of a single axis or all the axes as before.  **out : *ndarray, optional***  Alternative output array in which to place the result. Must be of the same shape and buffer length as the expected output. See **doc.ufuncs** (Section “Output arguments”) for more details.  **keepdims : *bool, optional***  If this is set to True, the axes which are reduced are left in the result as dimensions with size one. With this option, the result will broadcast correctly against the input array.  If the default value is passed, then *keepdims* will not be passed through to the [**amax**](https://docs.scipy.org/doc/numpy/reference/generated/numpy.amax.html#numpy.amax) method of sub-classes of [**ndarray**](https://docs.scipy.org/doc/numpy/reference/generated/numpy.ndarray.html#numpy.ndarray), however any non-default value will be. If the sub-class’ method does not implement *keepdims* any exceptions will be raised.  **initial : *scalar, optional***  The minimum value of an output element. Must be present to allow computation on empty slice. See [**reduce**](https://docs.scipy.org/doc/numpy/reference/generated/numpy.ufunc.reduce.html#numpy.ufunc.reduce) for details.  *New in version 1.15.0.*  **where : *array\_like of bool, optional***  Elements to compare for the maximum. See [**reduce**](https://docs.scipy.org/doc/numpy/reference/generated/numpy.ufunc.reduce.html#numpy.ufunc.reduce) for details.  *New in version 1.17.0.* |

print(np**.**where(a **<** 4, **-**1, 100))

where(condition, if true this value , false this value)

<https://www.pluralsight.com/guides/web-scraping-with-beautiful-soup>

print(emp1.\_\_dic\_\_)

'''emp\_1 = Employee('Divya', 'bandari', 66500, 'computer science')  
emp\_2 = Employee('Harsha', 'Maddhula ', 62400, 'computer science')  
  
print(emp\_1)  
print(emp\_2.name)  
print(Employee.data\_member)  
print("harsha")  
print(Employee.total\_salary)  
emp\_1.averageSalary()'''  
  
# print(Employee.averageSalary())