

Spring 2022: CSEE5590/490 – Special Topics

Python and Deep Learning - ICP-3

Lesson Overview:

In this lesson, we will review Object Oriented Python.

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 levels of data encapsulation like private, protected, and public.

Use Case Description:

1. Inheritance (Vehicle)
2. Web scraping

Programming elements:

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

1. Classes and objects:

Create a class Employee and then do the following:

- a. Create a constructor to initialize:
 - i. id, name, department, salary, balance, and isEmployed=True.
- b. Create a class attribute to count the number of Employees.
- c. Create a Full_time and Part_time Employee classes and it should inherit the properties of the Employee class.
 - i. Call init method of the parent class to create an object in these classes.
- d. Create giveRaise method for Full_time and Part_time classes. The full_time emp. Should have a default value of 10% increase and Part_time should have a default value of 5% increase.
- e. Create the following functions:
 - i. Pay: This should pay the employee once. Their balance should increase by the amount they are paid.
 - ii. Fire: this should remove the employee from the payroll. It should set their payrate to **0**, and isEmployed to false.
 1. (Note: Retain the records for the employee, just make sure they are not paid anymore.)
 - iii. isEmployed: a Boolean function that should return whether they are employed or not.
- f. Read employee data from **input.txt** to create instances of the Employee class:
 - i. File structure:
 1. **NEW**: keyword to create a new employee following: ID, Name, Department, Salary, Type(Full_time “F”, Part_time “P”).
 2. **RAISE**: keyword to give raise to a specific employee following: ID, raise_percentage that will change the emp. salary.
 3. **PAY**: keyword to pay all employee once, balance should increase by the amount they are paid (salary).
 4. **FIRE**: Keyword to remove the employee from payroll. Emp. Following is Emp. ID
- g. Create a function to find the average salary paid to all employees of each class. Write the average salary paid to the end of output.txt file after printing the total number of employees.
- h. Your program should output data to output.txt with the following format:
 - i. Emp. Name, ID ###, Department:

- ii. If employed, write out their Salary in this format: Current salary: \$##
- iii. If NOT employed, you should write out: Not employed with the company.
- iv. The employee's balance to date: Pay earned to date: \$##
- v. Full-Time or part-time employee.
- vi. Add a blank line between employees.
- vii.
- viii. Total number of employees: ###
- ix. Average Salary paid to all Full-time employees: \$###
- x. Average Salary paid to all Part-time employees: \$###

1.2 OPTIONAL Question:

- **Create a Car class with the following attributes:**
 - o **Private:**
 - Year
 - Make
 - Model
 - o **Init method to set (yea, make, model)**
 - o **Private:**
 - **Printinfo():**
 - **To print the object information (Make, Model, Year).**
 - o **Create an object and print its information using Printinfo().**

2. Web scraping

1. Write a simple program that parse a Wiki page mentioned below and follow the instructions:
 - https://en.wikipedia.org/wiki/Machine_learning
2. Print out the title of the page
3. Find all the images links in the page ('???' tag)
4. Iterate over each tag(above) then return the link using attribute "???" using get method.

**** Follow the IPC rubric guidelines**

Submission Guidelines:

1. Once finished document your code and make sure all parts if the assignments are completed.
2. Push your code to your GitHub repo and update the ReadMe file, add your info, and partner info.
3. Submit the assignment ICP-2 on Canvas.
4. Present your work to TA during class time to prove the execution and complete submission.

After class submission:

1. Once finished document your code and make sure all parts if the assignments are completed.
2. Push your code to your GitHub repo and update the ReadMe file, add your info, and partner info.
3. Submit the assignment ICP-2 on Canvas before the deadline.
4. Record a short video (1~3) minute, proof of execution and complete assignment.
5. Add video link to ReadMe file.

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-notice/academic-honesty/>