CIS 5357 Fall 2020

Programming Assignment # 9

(35 points)

**Due Date: Before 11:59 pm on Sunday, November , 22, 2020**

**Caution:**

**The instructor has found many submissions with duplicate or similar code. The instructor expects individual effort on each and every submission. Assignments and exams assigned in this course are neither group projects nor any kind of group activity or collaboration are sanctioned. Such activities will be treated per the Academic Dishonesty policy as stated in the course syllabus. Each submission will be closely examined for plagiarism**. **A slightest hint of duplicate or similar code will be examined for academic integrity violation. Such submissions will automatically receive a grade of zero and reported to the Graduate College for further disciplinary action. To avoid such scrutiny, please do your own work. Please consult your instructor if you any questions or need clarification about assignments and examinations.**

1. **Requirements for Assignment 9:**
2. Name your Jupyter Notebook ‘YourName-Assignment9.ipynb’
3. Include your name and submission date as level 2 headings in the first cell of the notebook.
4. Insert a markdown cell with level 3 heading - Grading Comments – Totality of all points noted below resulted in a reduction of xx Points.
5. Insert a markdown cell with level 3 heading for the Program Name and its objective
6. The program design must be broken down into functions, with each function performing a specified task. The main driver program will also be coded as main() function.
7. Each function must include an appropriately written docString.
8. Include functions defined for each program in their individual cells.
9. Main() function should be the last one defined.
10. To execute the Main() function, include a call statement in its own cell below the Main() function cell.
11. **Any data files used in the program should be accessed from the “/users/cis\_developer/CIS5357Fall2020/” location.**
12. Upload your source code file (.ipynb) to the Assignments section of Canvas ***BEFORE*** 11:59 pm on Sunday, November 22, 2020. **No other files need to be uploaded.**
13. **NO LATE ASSIGNMENTS WILL BE ACCEPTED. ASSIGNMENTS SENT VIA EMAIL AS ATTACHMENTS WILL ALSO NOT BE ACCEPTED. YOU ARE ALLOWED A MAXIMUM OF THREE ATTEMPTS TO SUBMIT YOUR FILE BEFORE THE DUE DEADLINE. YOU MISS THE DEADLINE, YOU LOSE IT. So, please start early to have a chance at getting any problem resolved before the submission deadline.**

**Introduction:**

Programming Assignment 9 will require you to design a python program that will use functions to modularize the program; consume JSON data retrieved via an Application Programming Interface or API or Endpoint (web service); consume CSV data, and use Pandas framework for data manipulation.

**Specifications:**

You will access a Web API (web service) to retrieve employee data for a fictitious company. You will use the following API URL to access the data:

<http://dummy.restapiexample.com/api/v1/employees>

The data returned by this URL is in JSON format and contains data on employee’s name, salary, age, and profile image. The value for profile image is currently empty and will not be used in this application. Your application will use Pandas framework for querying and manipulating this data.

**Requirements:**

1. **The program design must be modularized**, delegating each specific major task to individual functions. For this assignment, you will decide on the number of functions; task performed by each function; sequence in which these functions are invoked; name of the functions and arguments/parameters that these functions will work with. Essentially, you will design and develop your own hierarchy and IPO charts. Your submission MUST have a MAIN() function that will drive the rest of the program. SUBMISSIONS WITHOUT MODULARIZATION FOR ALL REQUIREMENTS WILL NOT BE GRADED.
2. Each function must include a docstring.
3. The program will NOT use ANY global variables. All variables must be local to either the function or main program. Functions and main program can exchange data only through the use of parameters in function calls and Return statements. Any libraries or packages to be imported will be imported inside the function that needs them.
4. When displaying tabulated output to the console, ensure that data are appropriate aligned within the columns and all columns are aligned/spaced to be visually appealing. Where possible and appropriate, use the Pandas dataframe formatting for displaying tabulated data.

**Process Requirements:**

1. Access the specified Web API and receive the JSON-formatted data into your application. The received data must eventually be stored in a Pandas dataframe. Name the dataframe ‘employee\_df.’
2. Once the data are received into a Pandas dataframe, you will manipulate the dataframe in following ways to generate some basic information.
   1. Modify the employee\_df contents to get them ready for further analysis.
      1. The profile image attribute is empty and will not be used in this application. Therefore, you should drop this from your employee\_df dataframe. That is, the employee\_df will be modified to not include this column for further analysis. Display the dataframe contents in close to demonstrate that the profile image column has indeed been removed from the dataframe.
      2. Next, if the datatype of values in the employee salary and age columns is of type string (object), convert the employee salary to floating point and age to integer data type. Make this change to employee\_df .
      3. Display the employee\_df to show that both modifications have been correctly made. Display the employee\_df dataframe and dataframe data types in console for verification.
   2. The data received from the Web API does not include employee’s home department, job classification (senior management or not), and gender. This data is available in a separate CSV file named “AdditionalData.csv.” Your application will read the data from this file into an appropriate data structure and then add these new attributes to the employee\_df dataframe modified in step 2(a). Now, your employee\_df will include employee’s name, salary, age, home department, job classification, and gender. Display the listing of all employee information in the employee\_df to the console.
   3. The company announces periodic bonus as a percentage of listed salary. All employees are given the same bonus percentage. Have your program accept the percentage of bonus from the user manager. Use 5% as a test value for the bonus percent. Next, apply this percentage to the employee’s current salary to compute the bonus such that a new column labeled ‘bonus’ is added to the employee\_df. Your dataframe should now include employee’s name, salary, age, department, job classification, gender, and bonus amount. Display a listing of all employee information in employee\_df after calculating the bonus to the console.
   4. Create a new dataframe by querying the employee\_df from step (2c) so that the new dataframe contains only employees making less than 100,000 in salary. Display employee name, salary, bonus, and gender. Name your new dataframe using a meaningful name representative of the contents.
   5. Create a new dataframe by filtering all employees younger than 40 years of age and assigned to the Client Services department. Display employee name, department and employee age in console Again, use a meaningful name for your new dataframe.
   6. Group the data in the employee\_df from step 2(c) by department and gender. Find the average salary by department and then gender. Display the average salary grouped by department first. Then display the average salary by gender. Looking at the results, do you think that there is a significant difference in average salary earned by male and female employees? Similarly, do you think that the average salary by department varies significantly? You may want to plot a bar charts comparing the average salary by gender and average salary by department.