Assignment 3

CS329e - Elements of Software Design
Company Employees
(100 points)
Due Date on Canvas and Gradescope

1 Description

A company has different types of employees and requires to manage the calculation of their salaries based on different parameters. Figure 1 illustrates the different types of employees of this company which are the following types: **Employee, Permanent_Employee, Temporary_Employee, Consultant, Manager and Consultant_Manager.**

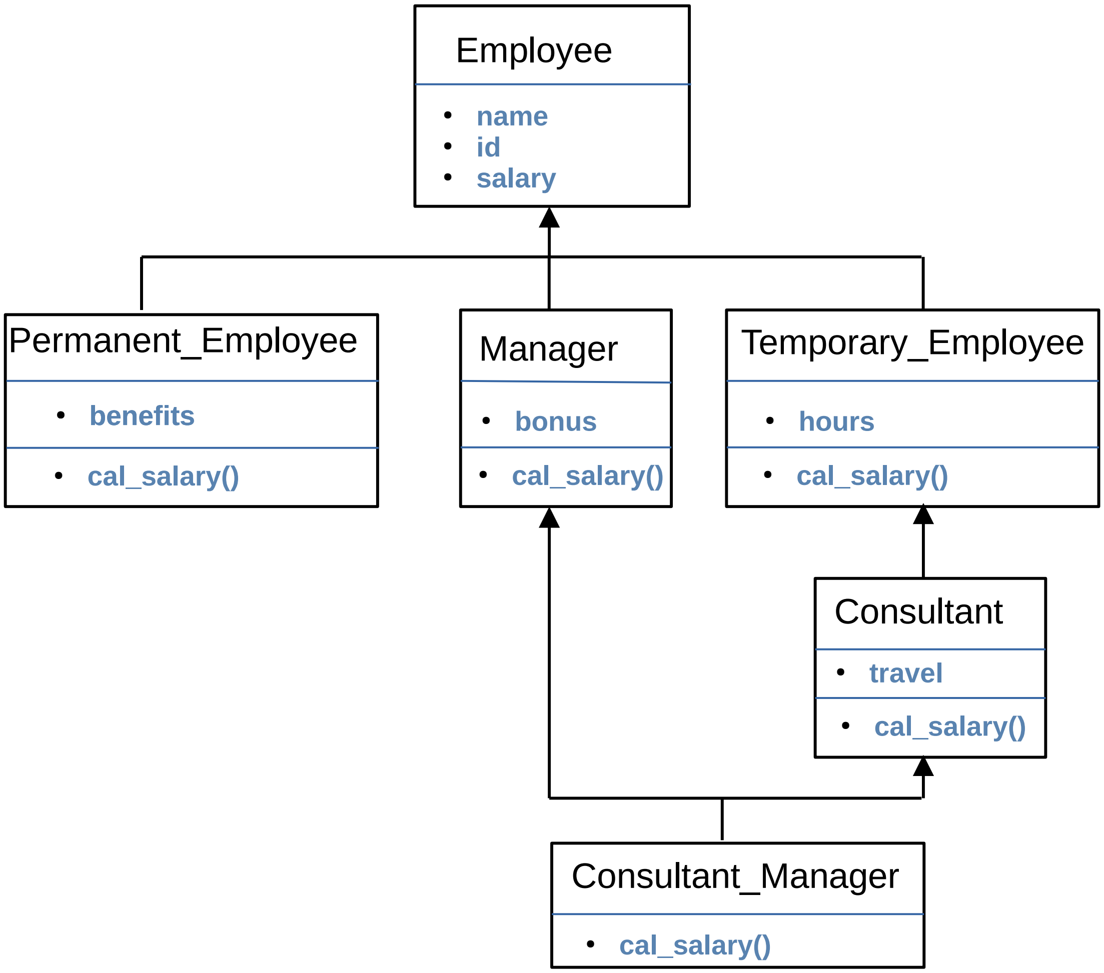

Figure 1: Class Diagram of Employees

- **Employee** has the following properties: name, id and salary.
- **Permanent_Employee** is a Employee. A Permanent Employee has benefits that he/she can select from [”health_insurance”], [”retirement”], or both [”retirement”, ”health_insurance”]. Implement a method **cal_salary()** to calculate the actual salary based on selected benefits. If benefits = [”health_insurance”] then return $salary * 0.9$, if [”retirement”] then $salary * 0.8$ and if benefits = [”retirement”, ”health_insurance”] is selected then $salary * 0.7$.
- **Temporary_Employee** has a property **”hours”** which is the working hours per mount. The salary property represents the amount that a temporary employee is paid per hour. Implement a method **cal_salary()** to calculate the actual salary by returning $salary * hours$
- **Consultant** a Consultant is a **Temporary_Employee** and has in addition to travel. The property travel represents number of travel trips that a Consultant has to do and is paid additionally 1000 dollar for each travel. Implement a method **cal_salary()** to calculate the actual salary similar to a Temporary_Employee with an additional payment of $travel * 1000$. (Tip: Try to reuse your code as much as possible to avoid having duplicated code)
- **Manager** A manager is a special type of employee and is paid with a bonus payment in addition to his/her main salary. Manager has a property bonus which adds to his/her main salary on top. Implement a method **cal_salary()** to return $salary + bonus$
- **Consultant_Manager** is a Consultant and is a Manager as well. A **Consultant_Manager** has travel, hours of work and bonus payment. Implement a method **cal_salary()** to calculate the salary similar to a Temporary_Employee and Consultant with an additional bonus payment. The cal_salary() method should return $salary * hours + travel * 1000 + bonus$

Also, please consider the following implementation instructions for this assignment:

- Each of your class constructors should use the following format **”__init__(self, **kwargs)”**
- Each of your classes should have a string method implementation (Define __str__(self) in each of your classes). You can design a string method to represent the objects in a string format. (print outs are not important for tests on Gradescope)
- You can find a template for your implementation in file **employee.py**. You do not need to change the main function. In main() function we will illustrate the various functions that you have written for the classes but it is not used to test your implementation.
- The file **(employee.py)** that you will be submitting should follow the standard coding conventions in Python[1] as good as possible. (We will not reduce points because of any violations)
- Write for each of your classes and methods a short text documentation.

[1] PEP 8 Style Guide for Python Code `https://www.python.org/dev/peps/pep-0008/`

How to run the template?

Mac: `python3 employee.py`

Windows: `python employee.py`

No input data is needed. Examples are implemented in main() function.

The example output will be:

Output:

```
Employee
Chris ,UT1,None

Permanent_Employee
Emma,UT2,100000,['health_insurance']

Temporary_Employee
Sam,UT3,100,40

Consultant
John ,UT4,100,40,10

Manager
Charlotte ,UT5,1000000,100000

Consultant_Manager
Matt ,UT6,1000,40,4,Consultant_Manager
Matt ,UT6,1000,10000

Check Salaries
Emma's Salary is: 90000.0

Emma's Salary is: 90000.0

Emma's Salary is: 70000.0

Sam's Salary is: 4000.0

John's Salary is: 14000.0

Charlotte's Salary is: 1100000.0

Matt's Salary is: 54000.0
```

You may not change the names of the functions listed. They must have the functionality as given in the specifications. You can always add more functions than those listed.

For this assignment you may work with a partner. Both of you must read the paper on Pair Programming[2] and abide by the ground rules as stated in that paper. If you are working with a partner then only one of you will be submitting the code. But make sure that your partner's name and UT EID is in the header. If you are

[2] Read this paper about Pair Programming `https://collaboration.csc.ncsu.edu/laurie/Papers/Kindergarten.PDF`

working alone then remove the partner's name and eid from the header.

1.1 Turnin

Turn in your assignment on time on Gradescope system on Canvas. For the due date of the assignments, please see the Gradescope and Canvas systems.

1.2 Academic Misconduct Regarding Programming

In a programming class like our class, there is sometimes a very fine line between ”cheating” and acceptable and beneficial interaction between students (In different assignment groups). Thus, it is very important that you fully understand what is and what is not allowed in terms of collaboration with your classmates. We want to be 100% precise, so that there can be no confusion.

The rule on collaboration and communication with your classmates is very simple: you cannot transmit or receive code from or to anyone in the class in any way – visually (by showing someone your code), electronically (by emailing, posting, or otherwise sending someone your code), verbally (by reading code to someone) or in any other way we have not yet imagined. Any other collaboration is acceptable.

The rule on collaboration and communication with people who are not your classmates (or your TAs or instructor) is also very simple: it is not allowed in any way, period. This disallows (for example) posting any questions of any nature to programming forums such as **StackOverflow**. As far as going to the web and using Google, we will apply the **”two line rule”**. Go to any web page you like and do any search that you like. But you cannot take more than two lines of code from an external resource and actually include it in your assignment in any form. Note that changing variable names or otherwise transforming or obfuscating code you found on the web does not render the ”two line rule” inapplicable. It is still a violation to obtain more than two lines of code from an external resource and turn it in, whatever you do to those two lines after you first obtain them.

Furthermore, you should cite your sources. Add a comment to your code that includes the URL(s) that you consulted when constructing your solution. This turns out to be very helpful when you're looking at something you wrote a while ago and you need to remind yourself what you were thinking.

We will use the following Code plagiarism Detection Software to automatically detect plagiarism.

- Staford MOSS

  `https://theory.stanford.edu/~aiken/moss/`

- Jplag - Detecting Software Plagiarism

  `https://github.com/jplag/jplag` and `https://jplag.ipd.kit.edu/`