Birzeit University
Department of Electrical & Computer Engineering
Second Semester, 2021/2022 ENCS313
Linux Laboratory
Python Project – University Employee Records Management System

Problem Overview:

In this project, you will manage the records of Birzeit University employees. This includes many operations such as adding a new employee record to the system, update some records, search for employee/s, and analyze data in records and present statistics. The employees' records are saved in text files. Each employee record contains several attributes. These attributes are divided into three categories; general attributes are those attributes for all employees, administrative attributes are those attributes for only administrative employees, and academic attributes are those attributes for only academic employees.

The general attributes are all required and are stored in a separate file with a structure similar to that of the attached GAttributes.txt file. The description of each of these attributes is:

- Employee ID: the ID of the employee. This attribute must contain only numbers with five digits and must be unique.
- Name: the name of the employee. This field must contain only characters.
- Date of birth,
- Marital status: this field has two possible values: single or married,
- Number of Childs,
- Gender: this field has two possible values: male or female,
- Contact information: {Email, Mobile number, Fax},
- Type: this field has two possible values: academic or administrative,
- Status: this field has three possible values: full-time, part-time, and left-university,
- Department: the place where the employee works for example computer engineering department,
- Starting time: the time when the employee started working at the university,
- Basic salary: the basic salary of the employee,
- Health insurance: true if the employee is enrolled with the university health insurance. Otherwise, it is false.

In this project, we will have only one administrative attribute that is the number of vacation days per year for the employee. This is described as the number of vacation days per year the employee took for each year the employee worked at Birzeit university. The value of this attribute must be saved in a dictionary type variable as below:

- Vacation = {**key**: year: **value** = number of days}.

The data for the vacation attribute is stored in a separate file with a structure similar to that of the attached Administrative.txt file.

We will also have only one academic attribute that is the academic experience of the employee. This is described as the courses the employee taught at Birzeit University. The value of this attribute must be saved in a dictionary type variable as below:

- Experience= {**key**: year-semester: **value** = list of all courses taught in that year-semester}.

The data for the experience attribute is stored in a separate file with a structure similar to that of the attached Academic.txt file.

Scenario:

- First, the program will ask to enter the name of the employee record file and raise an error if the file didn't exist.
- Then, the program will print the available list of options as following:
    1. **Add a new employee record**: here the program must ask to enter new record for an employee that consists of all general attributes only. The program must check the entered data.
    2. **Update general attributes:** here the system must ask for employee ID and then the name of the general attribute/s and finally the new value. The program must raise an error if the ID does not exist and must also check the entered data.
    3. **Add/update administrative employee attribute**: here the program must
        a. Ask for the employee ID and verify that the type of employee is administrative and the status not left-university. Otherwise return an error message and return to main menu.
        b. For administrative employees, the program then asks for the data of the vacation attribute including the year and number of vacation days. If the year is already in the employee Vacation attribute, the project should update the number of days. Otherwise, it will add a new {key: value} to the Vacation attribute.
    4. **Add/update academic employee attribute**: here the program must
        c. Ask for the employee ID and verify that the type of employee is academic and the status not left-university. Otherwise return an error message and return to main menu.
        d. For academic employees, the program then asks for the data of the Experience attribute including the year, semester, and courses. If the year and semester are already in the employee Experience attribute, the project should update the list of taught courses in that semester. Otherwise, it will add a new {key: value} to the Experience attribute.
    5. **Employee's statistics:** the program must display the following information:
        - Number of academic employees,
        - Number of administrative employees,
        - Percent of Full-time employees,
        - Number of Male employees,
        - Number of Female employees,
    6. **Salary statistics:** the program must display the following information:
        - Average academic employees' salary
        - Average administrative employees' salary

- The names (full name) of the employees who have salaries greater than a user defined number. Here the program must ask user to enter this number. Note that the final salary is computed as:
  Final salary = basic salary + 20 if the marital status is married + 15*number of Childs – 12 * (1 + (1 + number of Childs) if marital status is marid and joint the health insurance)

7. **Retirement information:** display the names (full name) of the employees whom have less than n remain years of service. You can compute this information using the staring time field given that maximum service years is 35 and from the date of birth filed given that retirement age is 65.

8. **Courses statistics:** the program must display for each course the following information:
   - Number of times the course is offered,
   - Number of academic employees who taught this course.

9. **Administrative employees' statistics:** the program must display for each administrative employee the following information:
   - Number of vacation days the employee took since the employee started working at Birzeit university,
   - The average number of vacation days per year for the employee. This is computed as the total number of vacation days divided by the number years the employee worked at Birzeit university.

10. **Academic employees' statistics:** the program must display for each academic employee the following information:
    - Number of courses the employee taught since the employee started working at Birzeit university,
    - The average number of courses the employee taught per semester. This is computed as the number of courses the employee taught at Birzeit university divided by the number of semesters the employee taught at Birzeit university.

Note that, after choosing and execute the option, the program will print again the list of options.

Submissions:

- You need to submit the code in .py format. If you made different files, you need to create a module and import them.
- Provide screenshots for each option
- Provide descriptions of the used data structures with clarifications (Why did you use it here?).

Important notes:

- Write the code for the Python program to satisfy the requirements described above.
- You must use functions (at least one function for each option).
- **You must use OOP concepts (classes, inheritance …). You must create a class for the general attributes and use inheritance to create classes for administrative and academic employees.**

- **You must organize your project in modules. I. e. have each class and the main function in separate modules (python scripts).**
- Make sure your code is clean and well indented; variables have meaningful names, etc.
- Make sure your script has enough comments inserted to add clarity.

Grading Policy:

Item	Weight
Read the name of the three input files and check if exist	5
Add a new employee record and check the validity of the entered data	10
Update general attributes and check the validity of the entered data	10
Add/update administrative employee attribute	10
Add/update academic employee attribute	10
Display employee statistic on screen	10
Salary statistics	10
Retainment information	5
Administrative employees' statistics	5
Academic employees' statistics	5
Handling options	5
Using OOP	15
Use modules	10
Code Structure	10
Run	20
Discussion	10
Total	**150**