Lesson – 5 – Inheritance and Polymorphism.

Day – 1 – Complete Q.No – 1 & 2

1. A student is a person, and so is an employee. Create a class Person that has the data attributes common to students and employees (name, social security number, age, gender, address, and telephone number) and necessary constructors, necessary getter and setters.

A student has a grade-point average (GPA), major, and year of graduation.

An employee has a department, job title, and date of hire. In addition, there are hourly employees (hourly rate, hours worked, and union dues) and salaried employees (annual salary).

Create a display() method to display the information's of each class attributes.

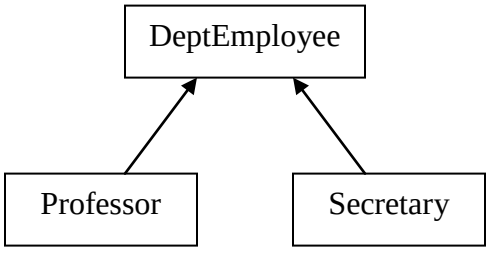
Define a class hierarchy and write an application class that you can use to first store the data for an array of people and then invoke display() the information in a meaningful way.

2. A university department consists of professors and secretaries. Each professor and each secretary has a name, a salary, and a hire date. Use inheritance and polymorphism to create an application that represents the department and its professors and secretaries as objects, and provides a test class that creates 3 professors and 2 secretaries, and then outputs the combined total of all of their salaries.

Start by creating classes

- Professor
- Secretary
- DeptEmployee

having the following relationship:



Place instance fields and corresponding accessor/mutator methods in DeptEmployee to represent name and hire date (as a type of LocalDate)(do not create accessors or mutators for salary). Do not put these fields in either the Professor or Secretary class. Also place in the Professor class an int field numberOfPublications, with corresponding accessor and mutator methods. Place in the Secretary class a double field overtimeHours, also with corresponding accessor and mutator methods.

Place a computeSalary method in DeptEmployee which simply returns the value stored in salary. Override the computeSalary method in Secretary so that the return value is the sum of the value of salary *plus* 12 times the number of overtime hours.

Then in the main method of a class named Main, create three instances of Professor and two instances of Secretary (you can invent the values to pass into the constructor). Finally, create an array of department employees:

```
DeptEmployee[] department = new DeptEmployee[5]
```

and then populate the array with the Professor and Secretary instances you have just created. Then ask the user if he wishes to see the sum of all Professor salary, sum of all secretary salary and all salaries in the department. If the user responds "Y", then loop through the department array and polymorphically read, and sum, all salaries, and output the result to the console.