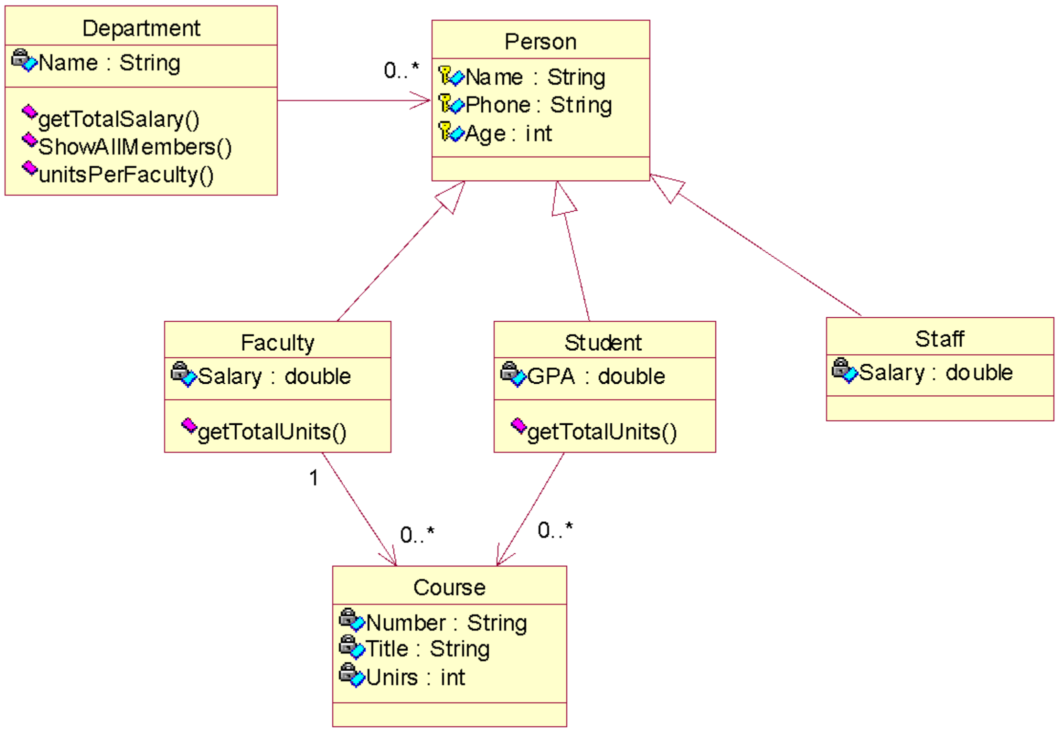
School Lab 2

Write a program that is used by the computer science department to keep track of students, faculty and staff within the department, the courses students take, and the courses faculty teach. The class diagram for this program is as follows:



A department has faculty, students and staff. Faculty teaches courses, and students take courses. The Department object has 3 methods:
double getTotalSalary() // computes the sum of all the salaries (per month) paid to faculty and staff.
void showAllMembers() // shows the name, phone number, age and type (student, faculty or staff) of all members in the department.
int unitsPerFaculty() // shows a list of all faculty names and the total number of units they teach.

Make sure your program reflects the structure of the class diagram shown above.
The class diagram does not show accessor functions for the attributes. Use the java 'ArrayList' class to implement a list of objects.

Level 1 – Implement the **getTotalSalary** functionality.
Level 2 – Implement the **showAllMembers** functionality.
Level 3 – Implement the **unitsPerFaculty** functionality.

Use the data below. Data should be hard coded into your application.

<u>**Level 4**</u>

Add the following feature to the lab :

Suppose staff members also want to take courses. They get a new title: staffstudents. They still get a salary, and they also have a GPA. For all staffstudents we want to keep track of the starting date of the first course they take. All existing functionality should still work, for example **getTotalSalary()** should also add salaries payed to staffstudents.

Please first show me your class diagram for the solution to this problem!

<u>**Level 5**</u>

Add the following feature :

The program asks for the name of a faculty member, and the program prints out the names of all students who take classes that are taught by this faculty member.

****

A department has faculty, students and staff. Faculty teaches courses, and students take courses.

<u>**You should use the skeleton program below. It does run.**</u>

Add the following data into your program to test it.(data is hard coded into your application)

Department: Computer Science
Faculty in the computer science department:

Name	Phone	Age	Salary
Frank Moore	472-5921	43	10000
Sam Howard	472-7222	55	9500
John Doodle	472-6190	39	8600

Students in the computer science department:

Name	Phone	Age	GPA
John Doe	472-1121	22	4.0
Mary Jones	472-7322	32	3.80
Lee Johnson	472-6009	19	3.65

Staff in the computer science department:

Name	Phone	Age	Salary
Frank Gore	472-3321	33	4050
Adam Davis	472-7552	50	5500
David heck	472-8890	29	3600

For level 3 up above :

Add the following data into your program to test it.(data is hard coded into your application)

Courses:

Number	Title	Units	Faculty name
CS201	Programming 1	4	John Doodle
CS360	Databases	3	Sam Howard
CS404	Compilers	4	John Doodle
CS240	Data structures	2	John Doodle
CS301	Software Engineering	3	Sam Howard
CS450	Advanced Architecture	5	Frank Moore

Courses per student:

Student name	Courses
John Doe	CS201, CS360, CS404, CS301
Mary Jones	CS201, CS404, CS450
Lee Johnson	CS201, CS360, CS240, CS450

Skeleton program :

```
import java.io.*;                  // for I/O
import java.lang.Integer;
import java.util.ArrayList;


/**
 * This is a skeleton class that you can use for guidance in starting the lab.
 *
 * You should update any comments,
 * variable names, etc to follow the class coding conventions.
 */

public class DepartmentApplication
   {
   public static void main(String[] args) throws IOException
      {
      Department dept = new Department("ComputerScience");

      //  The following commented out code will help you
      //  create the objects that you need.

      /*****************************************************

             // Create faculty objects
             Faculty frankMoore = new Faculty("Frank
Moore","472-5921",43,10000);
             Faculty samHoward = new Faculty("Sam Howard","472-7222",55,9500);
             Faculty johnDoodle = new Faculty("John Doodle","472-6190",39,8600);
             dept.addFaculty(frankMoore);
             dept.addFaculty(samHoward);
             dept.addFaculty(johnDoodle);


             // Create student objects
             Student johnDoe = new Student("John Doe","472-1121",22,4.0);
             Student maryJones = new Student("Mary Jones","472-7322",32,3.80);
             Student leeJohnson = new Student("Lee Johnson","472-6009",19,3.65);
             dept.addStudent(johnDoe);
             dept.addStudent(maryJones);
             dept.addStudent(leeJohnson);


             // Create staff objects
             Staff frankGore = new Staff("Frank Gore","472-3321",33,4050);
             Staff adamDavis = new Staff("Adam Davis","472-7552",50,5500);
             Staff davidHeck = new Staff("David Heck","472-8890",29,3600);
             dept.addStaff(frankGore);
             dept.addStaff(adamDavis);
             dept.addStaff(davidHeck);

             // Create course objects
             Course cs201 = new Course("cs201","programming",4, johnDoodle);
             Course cs360 = new Course("cs360","database",3, samHoward);
             Course cs404 = new Course("cs404","compiler",4, johnDoodle);
             Course cs240 = new Course("cs240","datastructure",2, johnDoodle);
             Course cs301 = new Course("cs301","Software engg",3, samHoward);
```

```
            Course cs450 = new Course("cs450","Advanced
architecture",5,frankMoore);

            /*
             * The above course objects will go inside either
             * a faculty object, or a student object.  Not all of the course
             *  objects go into the same object.
             *
             *  You would have code that looks something like :
             *  frankMoore.addCourse(cs450);
             *
             *  The addCourse method would have to be written in
             *  the faculty class.  Something similar would be done
             *  for students.
             */

      /*******************************************************/


      double totsalary = 0;

      while(true)
         {
         putText("Enter first letter of ");
         putText("getTotalSalary, showAllMembers, unitsPerFaculty or quit : ");
         int choice = getChar();
         switch(choice)
            {
            case 'g':
               totsalary=dept.getTotalSalary();
               putText("Total sum of all salaries is:");
               putText(String.valueOf(totsalary)+"\n");
               break;
            case 's':
               dept.showAllMembers();
               break;
            case 'u':
               dept.unitsPerFaculty();
               break;
            case 'q': return;
            default:
               putText("Invalid entry\n");
            }  // end switch
         }  // end while
      }  // end main()
// -------------------------------------------------------------
   public static void putText(String s) //writes string s to the screen
      {
      System.out.println(s);
      }
// -------------------------------------------------------------
   public static String getString() throws IOException  //reads a string from
the keyboard input
      {
      InputStreamReader isr = new InputStreamReader(System.in);
      BufferedReader br = new BufferedReader(isr);
      String s = br.readLine();
      return s;
      }
// ------------------------------------------------------------
   public static char getChar() throws IOException //reads a character from the
keyboard input
      {
      String s = getString();
      return s.charAt(0);
      }

//-------------------------------------------------------------
   public static int getInt() throws IOException // reads an integers from the
keyboard input
      {
      String s = getString();
      return Integer.parseInt(s);
      }
// ------------------------------------------------------------
   }  // end class
```

Good Luck!