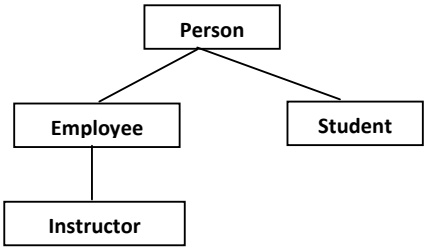
Questions

DCS-JMI wants you write some classes for their Personnel Record System. To make it simple, consider only 4 classes: Person, Employee, Instructor and Student. The following figure illustrates the relationship between these 4 classes. The Person class is the parent class of the Employee class and the Student class. The Employee class is the parent class of the Instructor class.



The following are the tasks you need to complete for each of these classes.

- Create appropriate fields for each class. Necessary fields are listed. Add your own fields if needed. Some fields need to have appropriate constraint. Use your own way to make sure that these constraints are satisfied.
  - **Person**
    - *ID*: int, starting from 1 and should be unique
    - *Name*: String
  - **Employee**
    - *Salary*: double and should not be negative
  - **Student** (For simplicity, assume that a student has at most 1 teacher)
    - *TeacherID*: int. It's his/her instructor's ID. 0 if no instructor is given
    - *TeacherName*: String
  - **Instructor**:
    - *StudentIDArray*: int array. An array of students' IDs of this instructor. Set the array size to be 10, initially all 0s, assuming an instructor won't have more than 10 students.
- All the above fields are private and only accessible through the access methods.
- A "toString()" method for each class to print out all the available information about the current object. In Person class "toString()" is declared as abstract.
- A static "findStudents(Person[] personArray)" method in the Instructor class to fill an instructor object's students ID array, and the corresponding students' TeacherID fields.
- Person should be declared as abstract class.
- Provide overloaded constructors/methods if needed.
- If a class can use the parent class method and constructor, use "super" to call it to reduce the redundant code.

Write a Test program to test the working of your classes in the main program. Also design the test-cases to test the program.